US009215225B2

(12) **United States Patent**
Desai

(10) **Patent No.: US 9,215,225 B2**
(45) **Date of Patent: Dec. 15, 2015**

(54) **MOBILE DEVICE LOCKING WITH CONTEXT**

(71) Applicant: **Citrix Systems, Inc.**, Fort Lauderdale, FL (US)

(72) Inventor: **Nitin Desai**, Coral Springs, FL (US)

(73) Assignee: **Citrix Systems, Inc.**, Fort Lauderdale, FL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 3 days.

(21) Appl. No.: **13/886,765**

(22) Filed: **May 3, 2013**

(65) **Prior Publication Data**

US 2014/0331285 A1 Nov. 6, 2014

(51) **Int. Cl.**
H04L 29/06 (2006.01)
G06F 21/60 (2013.01)

(52) **U.S. Cl.**
CPC ***H04L 63/08*** (2013.01); ***G06F 21/60*** (2013.01); ***H04L 63/1408*** (2013.01); *G06F 2221/2111* (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,805,803	A	9/1998	Birrell et al.
6,151,606	A	11/2000	Mendez
6,154,172	A	11/2000	Piccionelli et al.
6,480,096	B1	11/2002	Gutman et al.
6,609,198	B1	8/2003	Wood et al.
6,621,766	B2	9/2003	Brewer et al.
6,751,738	B2	6/2004	Wesinger, Jr. et al.
6,859,879	B2	2/2005	Henn et al.
6,883,098	B1	4/2005	Roman et al.
7,043,453	B2	5/2006	Stefik et al.
7,065,652	B1	6/2006	Xu et al.
7,159,120	B2	1/2007	Muratov et al.
7,240,015	B1	7/2007	Karmouch et al.
7,254,831	B2	8/2007	Saunders et al.
7,269,605	B1	9/2007	Nguyen et al.
7,340,772	B2	3/2008	Panasyuk et al.
7,415,498	B2	8/2008	Russo et al.

(Continued)

FOREIGN PATENT DOCUMENTS

EP	1465039	A1	10/2004
EP	2403211	A1	1/2012

(Continued)

OTHER PUBLICATIONS

Apple Inc., iPad User Guide for iOS 6.1 Software, Jan. 2013, Chapter 26, Accessibility, pp. 107-108.

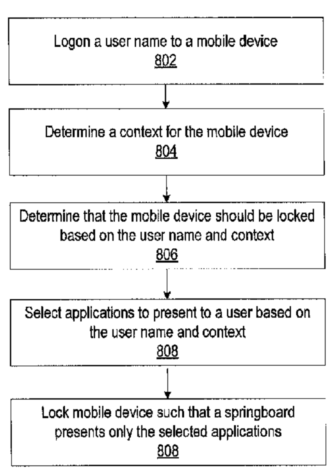
(Continued)

Primary Examiner — David Le
(74) *Attorney, Agent, or Firm* — Banner & Witcoff, Ltd.

(57) **ABSTRACT**

A method and system for locking a mobile device on an interface are described. A user logs on to a mobile device with a user name. The mobile device then determines a context for the mobile device based on one or more operational parameters and/or the user name. For example, a context for the mobile device may be a current location of the device. Based on the context and user name, the mobile device may run in locked mode. In locked mode, applications are selected to be presented on the mobile device based on the user name and context. The mobile device is locked on a springboard that presents only the selected applications to the user for launching. A user may switch between launched applications on the mobile device, but the user may only switch between launched applications that are presented on the springboard.

20 Claims, 12 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

7,437,752	B2	10/2008	Heard et al.
7,490,073	B1	2/2009	Qureshi et al.
7,490,352	B2	2/2009	Kramer et al.
7,496,954	B1	2/2009	Himawan et al.
7,502,861	B1	3/2009	Protassov et al.
7,509,672	B1	3/2009	Horwitz et al.
7,526,800	B2	4/2009	Wright et al.
7,529,923	B2	5/2009	Chartrand et al.
7,596,593	B2	9/2009	Mitchell et al.
7,599,991	B2	10/2009	Vargas et al.
7,665,125	B2	2/2010	Heard et al.
7,697,737	B2	4/2010	Aull et al.
7,716,240	B2	5/2010	Lim
7,761,523	B2	7/2010	May et al.
7,774,323	B2	8/2010	Helfman
7,779,408	B1	8/2010	Papineau
7,779,458	B1	8/2010	Heiderscheit et al.
7,788,535	B2	8/2010	Bussa et al.
7,788,536	B1	8/2010	Qureshi et al.
7,865,888	B1	1/2011	Qureshi et al.
7,904,468	B2	3/2011	Neil et al.
7,950,066	B1	5/2011	Zuili
7,966,323	B2	6/2011	Bocking et al.
7,966,652	B2	6/2011	Ganesan
7,970,386	B2	6/2011	Bhat et al.
7,970,923	B2	6/2011	Pedersen et al.
8,001,278	B2	8/2011	Huggahalli et al.
8,012,219	B2	9/2011	Mendez et al.
8,037,421	B2	10/2011	Scott et al.
8,051,180	B2	11/2011	Mazzaferri et al.
8,060,074	B2	11/2011	Danford et al.
8,060,596	B1	11/2011	Wootton et al.
8,078,713	B1	12/2011	Kim
8,085,891	B2	12/2011	Owen
8,095,517	B2	1/2012	Sandoval et al.
8,095,786	B1	1/2012	Kshirsagar et al.
8,126,128	B1	2/2012	Hicks, III et al.
8,126,506	B2	2/2012	Roundtree
8,181,010	B1	5/2012	Uchil et al.
8,200,626	B1	6/2012	Katzer et al.
8,214,887	B2	7/2012	Clark et al.
8,238,256	B2	8/2012	Nugent
8,239,918	B1	8/2012	Cohen
8,245,285	B1	8/2012	Ravishankar et al.
8,272,030	B1	9/2012	Annan et al.
8,285,681	B2	10/2012	Prahlad et al.
8,296,239	B2	10/2012	Nonaka
8,296,821	B2	10/2012	Nakae
8,332,464	B2	12/2012	Dispensa et al.
8,359,016	B2	1/2013	Lindeman et al.
8,365,258	B2	1/2013	Dispensa
8,365,266	B2	1/2013	Bogner
8,402,011	B1	3/2013	Bodenhamer
8,406,748	B2	3/2013	Raleigh et al.
8,418,238	B2	4/2013	Platt et al.
8,463,253	B2	6/2013	Chipalkatti et al.
8,463,946	B2	6/2013	Ferguson et al.
8,468,090	B2	6/2013	Lesandro et al.
8,468,455	B2	6/2013	Jorgensen et al.
8,495,746	B2	7/2013	Fissel et al.
8,528,059	B1	9/2013	Labana et al.
8,549,656	B2	10/2013	Blaisdell et al.
8,560,709	B1	10/2013	Shokhor et al.
8,578,443	B2	11/2013	Narain et al.
8,584,114	B2	11/2013	Rabinovich et al.
8,601,562	B2	12/2013	Milas
8,613,070	B1	12/2013	Borzycki et al.
8,650,303	B1	2/2014	Lang et al.
8,650,620	B2	2/2014	Chawla et al.
8,660,530	B2	2/2014	Sharp et al.
8,719,898	B1	5/2014	Barton et al.
8,799,994	B2	8/2014	Barton et al.
8,806,570	B2	8/2014	Barton et al.
8,843,734	B2	9/2014	Lim
8,850,010	B1	9/2014	Qureshi
8,850,049	B1	9/2014	Qureshi
8,856,909	B1	10/2014	Chickering
8,863,297	B2	10/2014	Sharma et al.
8,863,298	B2	10/2014	Akella et al.
8,863,299	B2	10/2014	Sharma et al.
8,881,228	B2	11/2014	Qureshi
8,918,834	B1	12/2014	Samuelsson
8,931,038	B2	1/2015	Pulier et al.
2001/0042045	A1	11/2001	Howard et al.
2002/0112047	A1	8/2002	Kushwaha et al.
2003/0031319	A1	2/2003	Abe et al.
2003/0037103	A1	2/2003	Salmi et al.
2003/0046366	A1	3/2003	Pardikar et al.
2003/0065947	A1	4/2003	Song et al.
2003/0131245	A1	7/2003	Linderman
2003/0157947	A1 *	8/2003	Fiatal et al. 455/466
2003/0229623	A1	12/2003	Chang et al.
2004/0006706	A1	1/2004	Erlingsson
2004/0010579	A1	1/2004	Freese
2004/0083273	A1	4/2004	Madison et al.
2004/0111640	A1	6/2004	Baum
2004/0117651	A1	6/2004	Little et al.
2004/0123153	A1	6/2004	Wright et al.
2004/0205233	A1	10/2004	Dunk
2004/0230807	A1	11/2004	Baird et al.
2005/0027843	A1	2/2005	Bozak et al.
2005/0055578	A1	3/2005	Wright et al.
2005/0076082	A1	4/2005	Le Pennec et al.
2005/0076085	A1	4/2005	Budd et al.
2005/0097608	A1	5/2005	Penke et al.
2005/0149340	A1	7/2005	Murakami et al.
2005/0172241	A1	8/2005	Daniels et al.
2005/0193222	A1	9/2005	Greene
2005/0255838	A1	11/2005	Adams et al.
2005/0262429	A1	11/2005	Elder et al.
2005/0265548	A1	12/2005	Tsuchimura
2005/0273592	A1	12/2005	Pryor et al.
2006/0005250	A1	1/2006	Chu et al.
2006/0070114	A1	3/2006	Wood et al.
2006/0075123	A1	4/2006	Burr et al.
2006/0085826	A1	4/2006	Funk et al.
2006/0094400	A1	5/2006	Beachem et al.
2006/0112428	A1 *	5/2006	Etelapera 726/16
2006/0117104	A1	6/2006	Taniguchi et al.
2006/0120526	A1	6/2006	Boucher et al.
2006/0141985	A1	6/2006	Patel et al.
2006/0147043	A1	7/2006	Mann et al.
2006/0161635	A1	7/2006	Lamkin et al.
2006/0185004	A1	8/2006	Song et al.
2006/0224742	A1	10/2006	Shahbazi
2006/0225142	A1	10/2006	Moon
2006/0242685	A1	10/2006	Heard et al.
2006/0259755	A1	11/2006	Kenoyer
2006/0282889	A1	12/2006	Brown et al.
2007/0005713	A1	1/2007	LeVasseur et al.
2007/0006327	A1	1/2007	Lal et al.
2007/0011749	A1	1/2007	Allison et al.
2007/0016771	A1	1/2007	Allison et al.
2007/0038764	A1	2/2007	Maillard
2007/0049297	A1	3/2007	Gopalan et al.
2007/0054627	A1	3/2007	Wormald
2007/0056043	A1	3/2007	Onyon et al.
2007/0072598	A1	3/2007	Coleman et al.
2007/0074033	A1	3/2007	Adams et al.
2007/0109983	A1	5/2007	Shankar et al.
2007/0118558	A1	5/2007	Kahandaliyanage
2007/0156897	A1	7/2007	Lim
2007/0180447	A1	8/2007	Mazzaferri et al.
2007/0186106	A1	8/2007	Ting et al.
2007/0198656	A1	8/2007	Mazzaferri et al.
2007/0199051	A1	8/2007	Parikh et al.
2007/0204153	A1	8/2007	Tome et al.
2007/0204166	A1	8/2007	Tome et al.
2007/0214272	A1	9/2007	Isaacson
2007/0226034	A1	9/2007	Khan
2007/0226225	A1	9/2007	Yiu et al.
2007/0226227	A1	9/2007	Helfman
2007/0226773	A1	9/2007	Pouliot
2007/0244987	A1	10/2007	Pedersen et al.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2007/0245409 A1	10/2007	Harris et al.	
2007/0248085 A1	10/2007	Volpano	
2007/0266422 A1	11/2007	Germano et al.	
2007/0283324 A1	12/2007	Geisinger	
2008/0027982 A1	1/2008	Subramanian et al.	
2008/0046580 A1	2/2008	Lafuente et al.	
2008/0047006 A1	2/2008	Jeong et al.	
2008/0047015 A1	2/2008	Cornwall et al.	
2008/0052395 A1	2/2008	Wright et al.	
2008/0066020 A1	3/2008	Boss et al.	
2008/0066177 A1	3/2008	Bender	
2008/0070495 A1	3/2008	Stricklen et al.	
2008/0092215 A1	4/2008	Soukup et al.	
2008/0127292 A1	5/2008	Cooper et al.	
2008/0133729 A1	6/2008	Fridman et al.	
2008/0134292 A1	6/2008	Ariel et al.	
2008/0141335 A1	6/2008	Thomas	
2008/0163188 A1	7/2008	Siskind et al.	
2008/0163286 A1	7/2008	Rudolph et al.	
2008/0194296 A1	8/2008	Roundtree	
2008/0196038 A1	8/2008	Antonio et al.	
2008/0196082 A1	8/2008	Sandoval et al.	
2008/0209506 A1	8/2008	Ghai et al.	
2008/0214300 A1	9/2008	Williams et al.	
2008/0229117 A1	9/2008	Shin et al.	
2008/0235760 A1	9/2008	Broussard et al.	
2008/0263224 A1	10/2008	Gilhuly et al.	
2008/0270240 A1	10/2008	Chu	
2008/0304665 A1	12/2008	Ma et al.	
2008/0313648 A1	12/2008	Wang et al.	
2008/0317292 A1	12/2008	Baker et al.	
2008/0318616 A1	12/2008	Chipalkatti et al.	
2009/0006232 A1	1/2009	Gallagher et al.	
2009/0028049 A1	1/2009	Boudreau et al.	
2009/0030968 A1	1/2009	Boudreau et al.	
2009/0037686 A1	2/2009	Mendonca	
2009/0037976 A1	2/2009	Teo et al.	
2009/0049425 A1	2/2009	Liepert et al.	
2009/0075630 A1	3/2009	McLean	
2009/0077638 A1	3/2009	Norman et al.	
2009/0083374 A1	3/2009	Saint Clair	
2009/0119773 A1	5/2009	D'Amore et al.	
2009/0121890 A1	5/2009	Brown et al.	
2009/0170532 A1	7/2009	Lee et al.	
2009/0172789 A1	7/2009	Band et al.	
2009/0178111 A1	7/2009	Moriconi et al.	
2009/0199178 A1	8/2009	Keller et al.	
2009/0199277 A1	8/2009	Norman et al.	
2009/0221278 A1	9/2009	Spelta et al.	
2009/0222880 A1	9/2009	Mayer et al.	
2009/0228714 A1	9/2009	Fiske et al.	
2009/0228954 A1	9/2009	Hu et al.	
2009/0228963 A1	9/2009	Pearce et al.	
2009/0249359 A1	10/2009	Caunter et al.	
2009/0253410 A1	10/2009	Fitzgerald et al.	
2009/0282127 A1	11/2009	Leblanc et al.	
2009/0282473 A1*	11/2009	Karlson et al.	726/17
2009/0323916 A1	12/2009	O'Sullivan et al.	
2009/0325615 A1	12/2009	McKay et al.	
2010/0064341 A1	3/2010	Aldera	
2010/0077469 A1	3/2010	Furman et al.	
2010/0100825 A1	4/2010	Sharoni	
2010/0100925 A1	4/2010	Hinton	
2010/0124196 A1	5/2010	Bonar et al.	
2010/0146523 A1	6/2010	Brigaut et al.	
2010/0146582 A1	6/2010	Jaber et al.	
2010/0150341 A1	6/2010	Dodgson et al.	
2010/0154025 A1	6/2010	Esteve Balducci et al.	
2010/0162232 A1	6/2010	Bhatia et al.	
2010/0173607 A1	7/2010	Thornton et al.	
2010/0180346 A1	7/2010	Nicolson et al.	
2010/0192212 A1	7/2010	Raleigh	
2010/0228825 A1	9/2010	Hegde et al.	
2010/0229197 A1	9/2010	Yi et al.	
2010/0248699 A1	9/2010	Dumais	
2010/0257580 A1	10/2010	Zhao et al.	
2010/0279652 A1	11/2010	Sharp et al.	
2010/0287619 A1	11/2010	Chase	
2010/0299152 A1	11/2010	Batchu et al.	
2010/0299376 A1	11/2010	Batchu et al.	
2010/0317336 A1	12/2010	Ferren et al.	
2010/0318992 A1	12/2010	Kushwaha et al.	
2010/0319053 A1	12/2010	Gharabally	
2010/0325097 A1	12/2010	Er et al.	
2010/0333165 A1	12/2010	Basak et al.	
2011/0030044 A1	2/2011	Kranendonk et al.	
2011/0072492 A1	3/2011	Mohler et al.	
2011/0145833 A1	6/2011	De Los Reyes et al.	
2011/0154266 A1	6/2011	Friend et al.	
2011/0154477 A1	6/2011	Parla et al.	
2011/0154498 A1	6/2011	Fissel et al.	
2011/0179484 A1	7/2011	Tuvell et al.	
2011/0208797 A1	8/2011	Kim	
2011/0208838 A1	8/2011	Thomas et al.	
2011/0209064 A1	8/2011	Jorgensen et al.	
2011/0209194 A1	8/2011	Kennedy	
2011/0219124 A1	9/2011	Allen et al.	
2011/0225417 A1	9/2011	Maharajh et al.	
2011/0239125 A1	9/2011	Kristensen et al.	
2011/0252232 A1	10/2011	De Atley et al.	
2011/0252459 A1	10/2011	Walsh et al.	
2011/0258301 A1	10/2011	McCormick et al.	
2011/0270963 A1	11/2011	Saito et al.	
2011/0271279 A1	11/2011	Pate	
2011/0276683 A1	11/2011	Goldschlag et al.	
2011/0276699 A1	11/2011	Pedersen	
2011/0277027 A1	11/2011	Hayton et al.	
2011/0283347 A1	11/2011	Bhuta et al.	
2011/0295970 A1	12/2011	Miyazawa	
2011/0314534 A1	12/2011	James	
2012/0002813 A1	1/2012	Wei et al.	
2012/0005476 A1	1/2012	Wei et al.	
2012/0005724 A1	1/2012	Lee	
2012/0005745 A1	1/2012	Wei et al.	
2012/0005746 A1	1/2012	Wei et al.	
2012/0023506 A1	1/2012	Maeckel et al.	
2012/0036347 A1	2/2012	Swanson et al.	
2012/0036370 A1	2/2012	Lim et al.	
2012/0042036 A1	2/2012	Lau et al.	
2012/0052954 A1	3/2012	Zhu et al.	
2012/0054853 A1	3/2012	Gupta et al.	
2012/0066691 A1	3/2012	Branton	
2012/0079475 A1	3/2012	Hicks, III et al.	
2012/0079556 A1	3/2012	Wahl	
2012/0084184 A1	4/2012	Raleigh et al.	
2012/0088540 A1	4/2012	Smith et al.	
2012/0096533 A1	4/2012	Boulos et al.	
2012/0096544 A1	4/2012	Hosoda	
2012/0102195 A1	4/2012	Adams et al.	
2012/0109384 A1	5/2012	Stepanian	
2012/0110317 A1	5/2012	Scheer et al.	
2012/0117622 A1	5/2012	Gronholm et al.	
2012/0129503 A1	5/2012	Lindeman et al.	
2012/0131116 A1	5/2012	Tu et al.	
2012/0131343 A1	5/2012	Choi et al.	
2012/0131685 A1	5/2012	Broch et al.	
2012/0151033 A1	6/2012	Baliga et al.	
2012/0154265 A1*	6/2012	Kim et al.	345/156
2012/0154413 A1	6/2012	Kim et al.	
2012/0157165 A1	6/2012	Kim et al.	
2012/0157166 A1	6/2012	Kim et al.	
2012/0159139 A1	6/2012	Kim et al.	
2012/0165075 A1	6/2012	Kim et al.	
2012/0166516 A1	6/2012	Simmons et al.	
2012/0166524 A1	6/2012	Watakabe et al.	
2012/0166997 A1	6/2012	Cho et al.	
2012/0167118 A1	6/2012	Pingili et al.	
2012/0167159 A1	6/2012	Mefford, Jr. et al.	
2012/0174237 A1*	7/2012	Krzyzanowski	726/29
2012/0179802 A1	7/2012	Narasimhan et al.	
2012/0179909 A1	7/2012	Sagi et al.	
2012/0185910 A1	7/2012	Miettinen et al.	
2012/0185913 A1	7/2012	Martinez et al.	
2012/0191716 A1	7/2012	Omoigui	

(56) **References Cited**

U.S. PATENT DOCUMENTS

2012/0198570	A1	8/2012	Joa et al.
2012/0204220	A1	8/2012	Lavi
2012/0210443	A1	8/2012	Blaisdell et al.
2012/0214472	A1	8/2012	Tadayon et al.
2012/0222120	A1	8/2012	Rim et al.
2012/0233130	A1	9/2012	Vedachalam et al.
2012/0238257	A1	9/2012	Anson
2012/0240183	A1	9/2012	Sinha
2012/0254768	A1	10/2012	Aggarwal et al.
2012/0255026	A1	10/2012	Baca et al.
2012/0265792	A1	10/2012	Salters
2012/0270522	A1	10/2012	Laudermilch et al.
2012/0272221	A1	10/2012	Pessoa et al.
2012/0278454	A1	11/2012	Stewart et al.
2012/0284325	A1	11/2012	Erb
2012/0284779	A1	11/2012	Ingrassia, Jr. et al.
2012/0290694	A9	11/2012	Marl et al.
2012/0291114	A1	11/2012	Poliashenko et al.
2012/0303476	A1	11/2012	Krzyzanowski et al.
2012/0303778	A1	11/2012	Ahiska et al.
2012/0304310	A1	11/2012	Blaisdell
2012/0311154	A1	12/2012	Morgan
2012/0311659	A1	12/2012	Narain et al.
2012/0317185	A1	12/2012	Shah et al.
2012/0321087	A1	12/2012	Fleischman et al.
2012/0324568	A1	12/2012	Wyatt et al.
2012/0331088	A1	12/2012	O'Hare et al.
2012/0331527	A1	12/2012	Walters et al.
2012/0331528	A1	12/2012	Fu et al.
2013/0002725	A1	1/2013	Kim et al.
2013/0007245	A1	1/2013	Malik et al.
2013/0007842	A1	1/2013	Park et al.
2013/0013653	A1	1/2013	Thompson
2013/0013688	A1	1/2013	Wang et al.
2013/0013932	A1	1/2013	Kong et al.
2013/0014239	A1	1/2013	Pieczul et al.
2013/0014267	A1	1/2013	Farrugia et al.
2013/0024424	A1	1/2013	Prahlad et al.
2013/0024928	A1	1/2013	Burke et al.
2013/0035063	A1	2/2013	Fisk et al.
2013/0042294	A1	2/2013	Colvin et al.
2013/0054922	A1	2/2013	Tuch et al.
2013/0054962	A1	2/2013	Chawla et al.
2013/0055378	A1	2/2013	Chang et al.
2013/0059284	A1	3/2013	Giedgowd, Jr. et al.
2013/0066960	A1	3/2013	Fieremans et al.
2013/0066978	A1	3/2013	Bentley et al.
2013/0067229	A1	3/2013	German et al.
2013/0074142	A1	3/2013	Brennan et al.
2013/0084847	A1	4/2013	Tibbitts et al.
2013/0086684	A1	4/2013	Mohler
2013/0091543	A1	4/2013	Wade et al.
2013/0097421	A1	4/2013	Lim
2013/0097660	A1	4/2013	Das et al.
2013/0111540	A1	5/2013	Sabin
2013/0117240	A1	5/2013	Taylor et al.
2013/0117563	A1	5/2013	Grabelkovsky
2013/0117805	A1	5/2013	Kent et al.
2013/0117840	A1	5/2013	Roesner et al.
2013/0124673	A1	5/2013	Hjelm et al.
2013/0130651	A1	5/2013	Deasy et al.
2013/0130652	A1	5/2013	Deasy et al.
2013/0130653	A1	5/2013	Deasy et al.
2013/0132457	A1	5/2013	Diwakar
2013/0132941	A1	5/2013	Lindeman et al.
2013/0133061	A1	5/2013	Fainkichen et al.
2013/0138766	A1	5/2013	Draluk et al.
2013/0138810	A1	5/2013	Binyamin et al.
2013/0139241	A1	5/2013	Leeder
2013/0142043	A1	6/2013	Tapia et al.
2013/0145448	A1*	6/2013	Newell 726/7
2013/0151598	A1	6/2013	Fu et al.
2013/0167247	A1	6/2013	Brown et al.
2013/0171967	A1	7/2013	Ashour et al.
2013/0212212	A1	8/2013	Addepalli et al.
2013/0219176	A1	8/2013	Akella et al.
2013/0219211	A1	8/2013	Gopinath et al.
2013/0219456	A1	8/2013	Sharma et al.
2013/0227636	A1	8/2013	Bettini et al.
2013/0227659	A1	8/2013	Raleigh
2013/0232541	A1	9/2013	Kapadia et al.
2013/0254262	A1	9/2013	Udall
2013/0254660	A1	9/2013	Fujioka
2013/0254831	A1	9/2013	Roach et al.
2013/0263208	A1	10/2013	Challa
2013/0263209	A1	10/2013	Panuganty
2013/0268676	A1	10/2013	Martins et al.
2013/0283335	A1	10/2013	Lakshminarayanan et al.
2013/0288656	A1	10/2013	Schultz et al.
2013/0290709	A1	10/2013	Muppidi et al.
2013/0291052	A1	10/2013	Hadar et al.
2013/0297604	A1	11/2013	Sutedja et al.
2013/0297662	A1	11/2013	Sharma et al.
2013/0298185	A1	11/2013	Koneru et al.
2013/0298201	A1	11/2013	Aravindakshan et al.
2013/0298242	A1	11/2013	Kumar et al.
2013/0303194	A1	11/2013	Rowles
2013/0311593	A1	11/2013	Prince et al.
2013/0311597	A1	11/2013	Arrouye et al.
2013/0318345	A1	11/2013	Hengeveld
2013/0333005	A1	12/2013	Kim et al.
2013/0346268	A1	12/2013	Pereira et al.
2014/0006347	A1	1/2014	Qureshi et al.
2014/0006512	A1	1/2014	Huang et al.
2014/0007183	A1	1/2014	Qureshi et al.
2014/0007214	A1	1/2014	Qureshi et al.
2014/0007215	A1	1/2014	Romano et al.
2014/0020062	A1	1/2014	Tumula et al.
2014/0020073	A1	1/2014	Ronda et al.
2014/0032691	A1	1/2014	Barton et al.
2014/0032733	A1	1/2014	Barton et al.
2014/0032758	A1	1/2014	Barton et al.
2014/0032759	A1	1/2014	Barton et al.
2014/0033271	A1	1/2014	Barton et al.
2014/0040638	A1	2/2014	Barton et al.
2014/0040656	A1	2/2014	Ho et al.
2014/0040979	A1	2/2014	Barton et al.
2014/0047535	A1	2/2014	Parla et al.
2014/0059640	A9	2/2014	Raleigh et al.
2014/0059642	A1	2/2014	Deasy et al.
2014/0096199	A1	4/2014	Dave et al.
2014/0108649	A1	4/2014	Barton et al.
2014/0130174	A1	5/2014	Celi, Jr. et al.
2014/0173700	A1	6/2014	Awan et al.
2014/0181934	A1	6/2014	Mayblum et al.
2014/0189808	A1	7/2014	Mahaffey et al.
2014/0298401	A1	10/2014	Batson et al.
2014/0315536	A1	10/2014	Chow et al.
2015/0026827	A1	1/2015	Kao et al.
2015/0087270	A1	3/2015	Richardson et al.

FOREIGN PATENT DOCUMENTS

EP	2428894	A1	3/2012
EP	2523107	A1	11/2012
GB	2411320	A	8/2005
GB	2462442	A	2/2010
WO	9914652	A1	3/1999
WO	02084460	A2	10/2002
WO	2004107646	A1	12/2004
WO	2007113709	A1	10/2007
WO	2008086611	A1	7/2008
WO	2009021200	A1	2/2009
WO	2010054258	A1	5/2010
WO	2010115289	A1	10/2010

OTHER PUBLICATIONS

Notice of Allowance issued in U.S. Appl. No. 14/022,845 mailed Dec. 6, 2013.

Notice of Allowance issued in corresponding U.S. Appl. No. 14/041,923, mailed Dec. 23, 2013.

Lowe, "Application-Specific VPNs," Dec. 13, 2005.

(56) **References Cited**

OTHER PUBLICATIONS

Restriction Requirement issued in U.S. Appl. No. 13/649,071 mailed Nov. 22, 2013.
International Search Report and Written Opinion mailed Nov. 26, 2013 in Internation Application No. PCT/US2013/060388.
Restriction Requirement issued in U.S. Appl. No. 13/649,076 mailed Jan. 13, 2013.
Xuetao Wei, et al., "Malicious Android Applications in the Enterprise: What Do They Do and How Do We Fix It?,"• ICDE Workshop on Secure Data Management on Smartphones and Mobiles, Apr. 2012, 4 pages.
Notification of Concurrently filed Applications in 1 page.
Ranjan et al., "Programming Cloud Resource Orchestration Framework: Operations and Research Challenges", arvix.org, 2012, pp. 1-19.
Na et al., "Personal Cloud Computing Security Framework," 2010 IEEE Asia-Pacific Computing Conference, 2010, pp. 671-675.
Notice of Allowance issued in U.S. Appl. No. 13/963,825 mailed Oct. 25, 2013.
Wilson et al., "Unified Security Framework", In proceedings of the 1st International Symposium on Information and Communication Technologies, pp. 500-505. Trinity College Dublin, 2003.
Mysore et al., "The Liquid Media System—a Multi-Device Streaming Media Orchestration Framework", Ubicomp 2003 Workshop, pp. 1-4.
Restriction Requirement issued in U.S. Appl. No. 13/963,833 mailed Dec. 19, 2013.
Written Opinion and International Search Report, PCT/US2013/062636, Jan. 10, 2014.
International Search Report and Written Opinion dated Feb. 4, 2014 in Application No. PCT/US2013/064349.
International Search Report and Written Opinion mailed Jan. 21, 2014 in International Application No. PCT/US2013/063856.
"Citrix XenMobile Technology Overview: White Paper," Citrix White Papers online, Jul. 31, 2012, pp. 1-14; retrieved from http://insight.com/content/aam/insight/en_US/pdfs/citrix/xenmobile-tech-overview.pdf, retrieved Jan. 27, 2014.
Wright et al., "Your Firm's Mobile Devices: How Secure are They?". Journal of Corporate Accounting and Finance. Jul. 1, 2011. Willey Periodicals. pp. 13-21.
Andreas, Digging into The Exchange ActiveSync Protocol, Mobility Dojo.net, Oct. 25, 2010, http://mobilitydoj.net/2010/03/17/digging-into-the-exchange-activesync-protocol/.
Feb. 12, 2015—(US) Non-Final Office Action—U.S. Appl. No. 13/963,833.
Feb. 18, 2015—(US) Non-Final Office Action—U.S. Appl. No. 14/043,229.
International Search Report and Written Opinion issued in PCT/US2013/063363, dated Dec. 20, 2013.
Dec. 19, 2014—(US) Non-Final Office Action—U.S. Appl. No. 14/043,331.
Mar. 5, 2015 (US) Non-Final Office Action—U.S. Appl. No. 14/039,651.
Apr. 2, 2015—(US) Final Office Action—U.S. Appl. No. 14/022,935.
Mar. 25, 2015—(US) Non-Final Office Action—U.S. Appl. No. 14/044,919.
Apr. 6, 2015—(US) Non-Final Office Action—U.S. Appl. No. 14/039,632.
Apr. 14, 2015—(US) Notice of Allowance & Fees Due—U.S. Appl. No. 14/043,086.
Written Opinion and International Search Report, PCT/US13/63261, Jul. 11, 2014.
May 19, 2015—(US) Notice of Allowance & Fees Due—U.S. Appl. No. 13/648,993.
May 22, 2015—(US) Non-Final Office Action—U.S. Appl. No. 14/106,171.
Laverty, Joseph Packy, et al., Comparative Analysis of Mobile Application Development and Security Models, [Online] 2011, Issues in Information Systems vol. XII, No. 1, [Retrieved from the Internet] <http://iacis.org/iis/2011/301-312_AL2011_1694.pdf> pp. 301-312.
Potharaju, Rahul, et al., Plagiarizing smartphone applications: attack strategies and defense techniques, [Online] 2012, Engineering Secure Software and Systems, Springer Berlin Heidelberg, [Retrieved from the Internet] <http://link.springer.com/chapter/10.1007/978-3-642-28166-2_11#> pp. 106-120.
Peine, H., Security concepts and implementation in the Ara mobile agent system, [Online] 1998, Enabling Technologies: Infrastructure for Collaborative Enterprises, 1998 Seventh IEEE International Workshops on Jun. 17-19, 1998, [Retrieved from the Internet] <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=725699&isnumber=15665> pp. 236-242.
Shah et al., Securing Java-Based Mobile Agents through Byte Code Obfuscation Techniques, [Online] Dec. 23-24, 2006, Multitopic Conference, 2006, INMIC '06. IEEE, [Retrieved from the Internet] <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4196425&isnumber=414013> pp. 305-308.
May 20, 2015—(US) Notice of Allowance & Fees Due—U.S. Appl. No. 13/649,022.
May 20, 2015—(US) Notice of Allowance & Fees Due—U.S. Appl. No. 13/649,069.
Jun. 5, 2015—(US) Final Office Action—U.S. Appl. No. 13/649,071.
Jun. 9, 2015—(US) Notice of Allowance—U.S. Appl. No. 14/043,229.
Jun. 15, 2015—(US) Non-Final Office Action—U.S. Appl. No. 13/649,076.
Jun. 18, 2015—(US) Non-Final Office Action—U.S. Appl. No. 14/043,902.
Jun. 23, 2015—(US) Notice of Allowance—U.S. Appl. No. 14/043,229.
No stated author; Administration Guide for Symantec Endpoint Protection and Symantec Network Access Control; 2008; Retrieved from the Internet <URL:ftp.symantec.com/public/english_us_canada/products/symantec_endpoint_protection/11.0/manals/administration_guide.pdf>; pp. 1-615 as printed.
No stated author; Symantec Network Access Control Enforcer Implementation Guide; 2007; Retrieved from the Internet <URL:ftp.symantec.com/public/english_us_canada/products/symantec_network_access_control/11.0/manuals/enforcer_implementation_guide.pdf>; pp. 1-132 as printed.
Jul. 6, 2015—(US) Notice of Allowance—U.S. Appl. No. 13/649,024.
Jul. 9, 2015—(US) Final Offcie Action—U.S. Appl. No. 13/963,833.

* cited by examiner

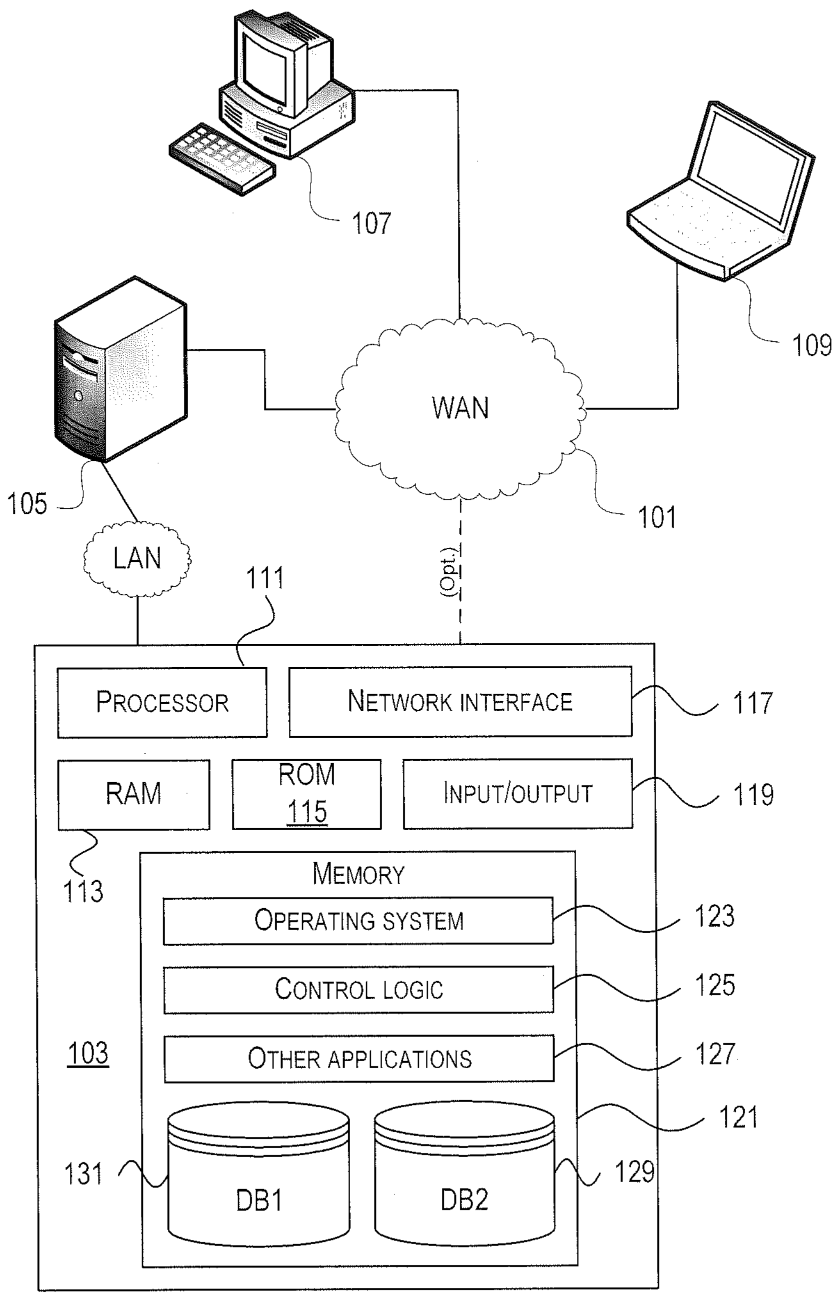

107
109
105
WAN
101
LAN
111
(Opt.)
PROCESSOR
NETWORK INTERFACE
117
RAM
ROM
115
INPUT/OUTPUT
119
113
MEMORY
OPERATING SYSTEM
123
CONTROL LOGIC
125
103
OTHER APPLICATIONS
127
121
131
DB1
DB2
129

FIG. 1

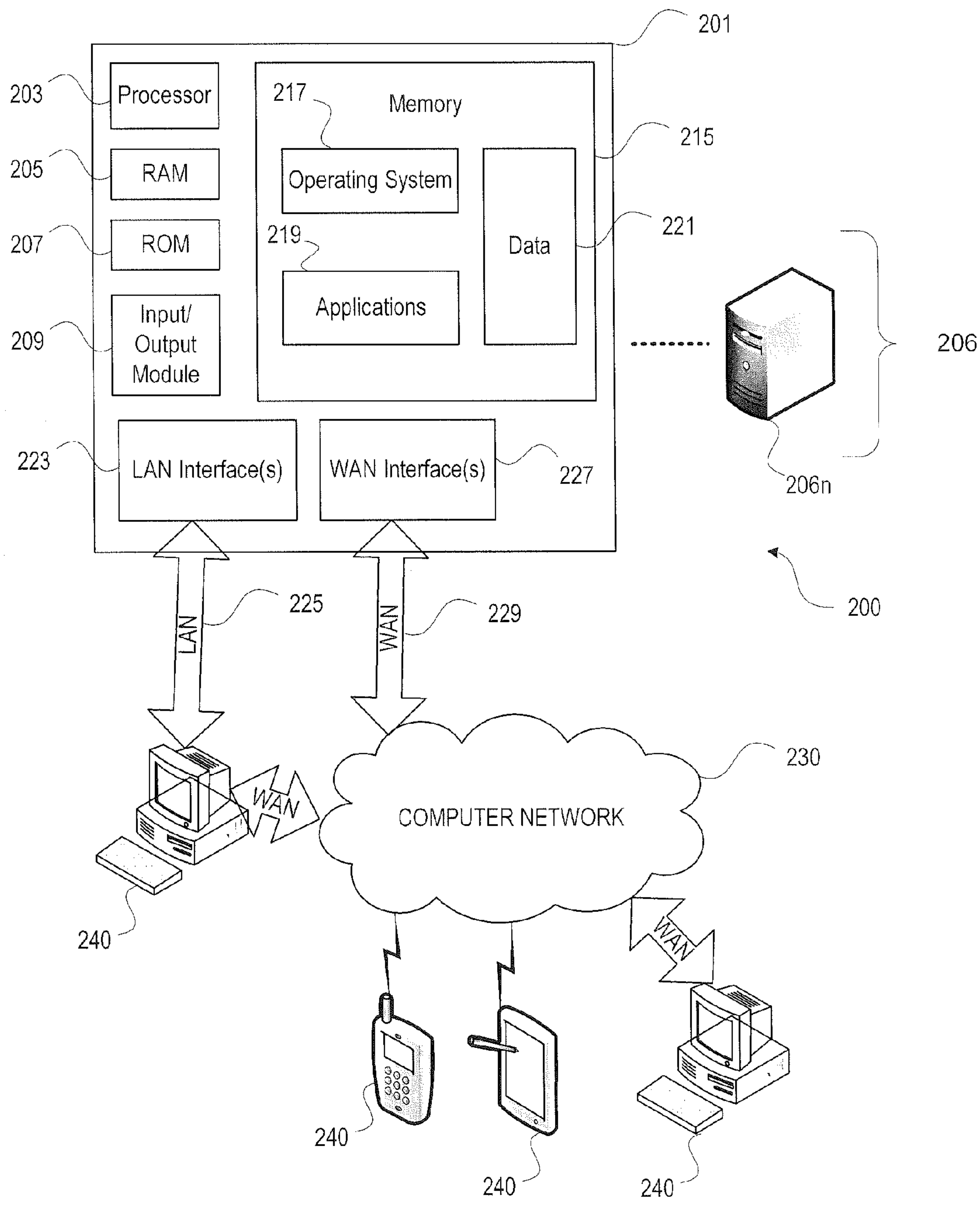

201
203
Processor
205
RAM
207
ROM
209
Input/
Output
Module
217
Memory
Operating System
219
Applications
Data
215
221
206
206n
223
LAN Interface(s)
WAN Interface(s)
227
LAN
225
WAN
229
200
WAN
COMPUTER NETWORK
230
WAN
240
240
240
240

FIG. 2

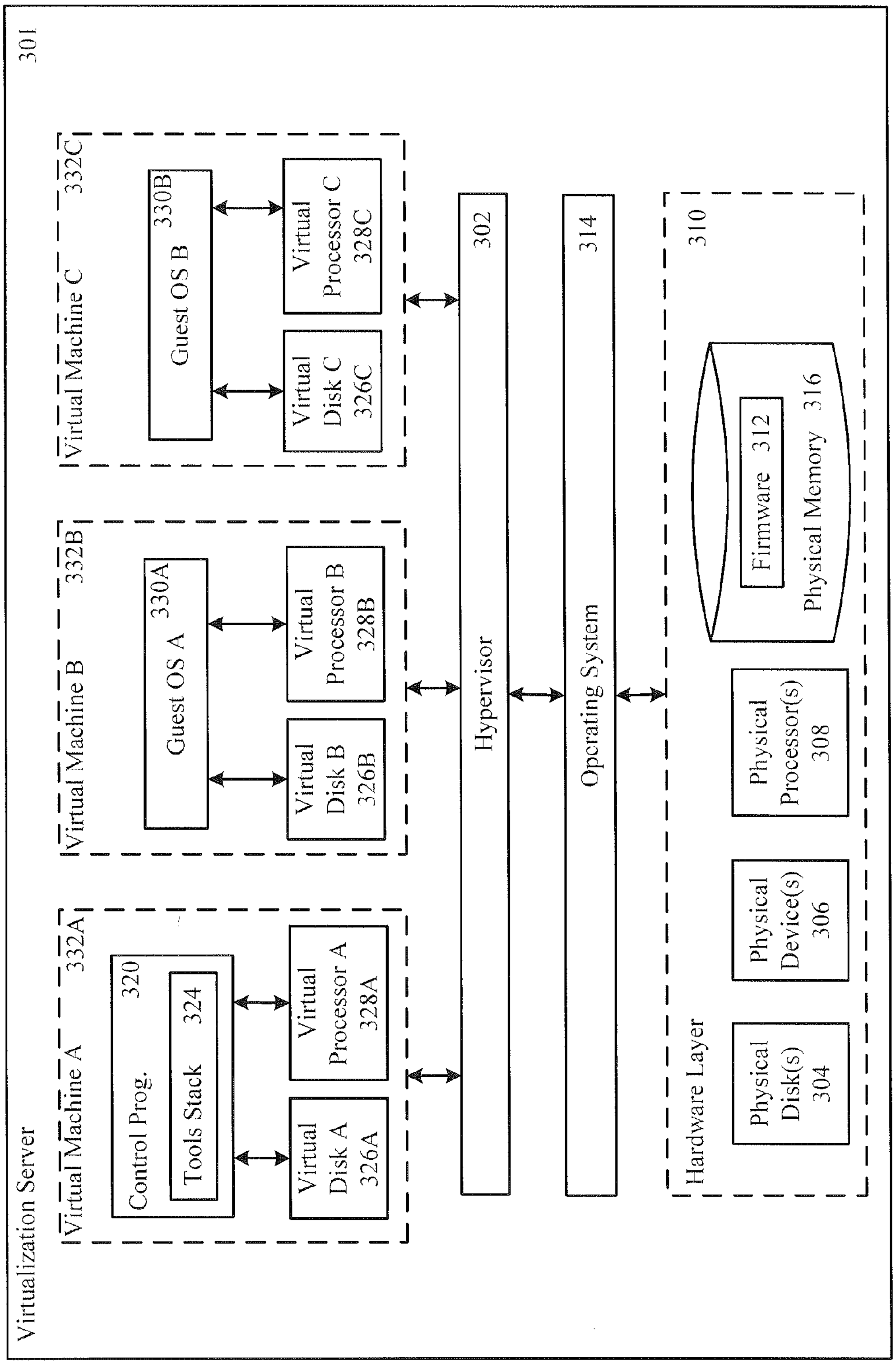

Virtualization Server
301
Virtual Machine A
332A
Control Prog.
320
Tools Stack
324
Virtual Disk A 326A
Virtual Processor A 328A
Virtual Machine B
332B
Guest OS A
330A
Virtual Disk B 326B
Virtual Processor B 328B
Virtual Machine C
332C
Guest OS B
330B
Virtual Disk C 326C
Virtual Processor C 328C
Hypervisor
302
Operating System
314
Hardware Layer
310
Physical Disk(s) 304
Physical Device(s) 306
Physical Processor(s) 308
Firmware 312
Physical Memory 316

FIG. 3

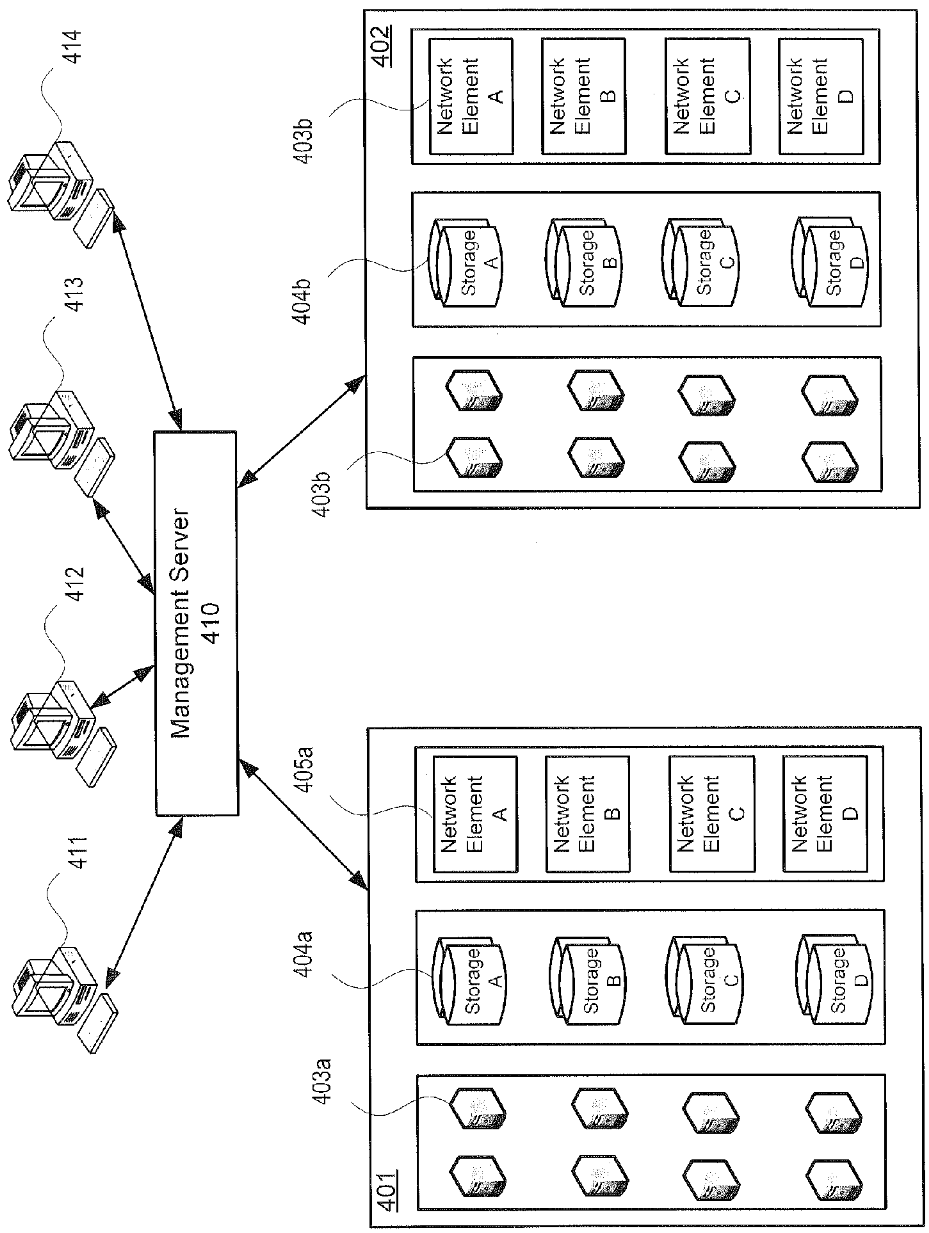
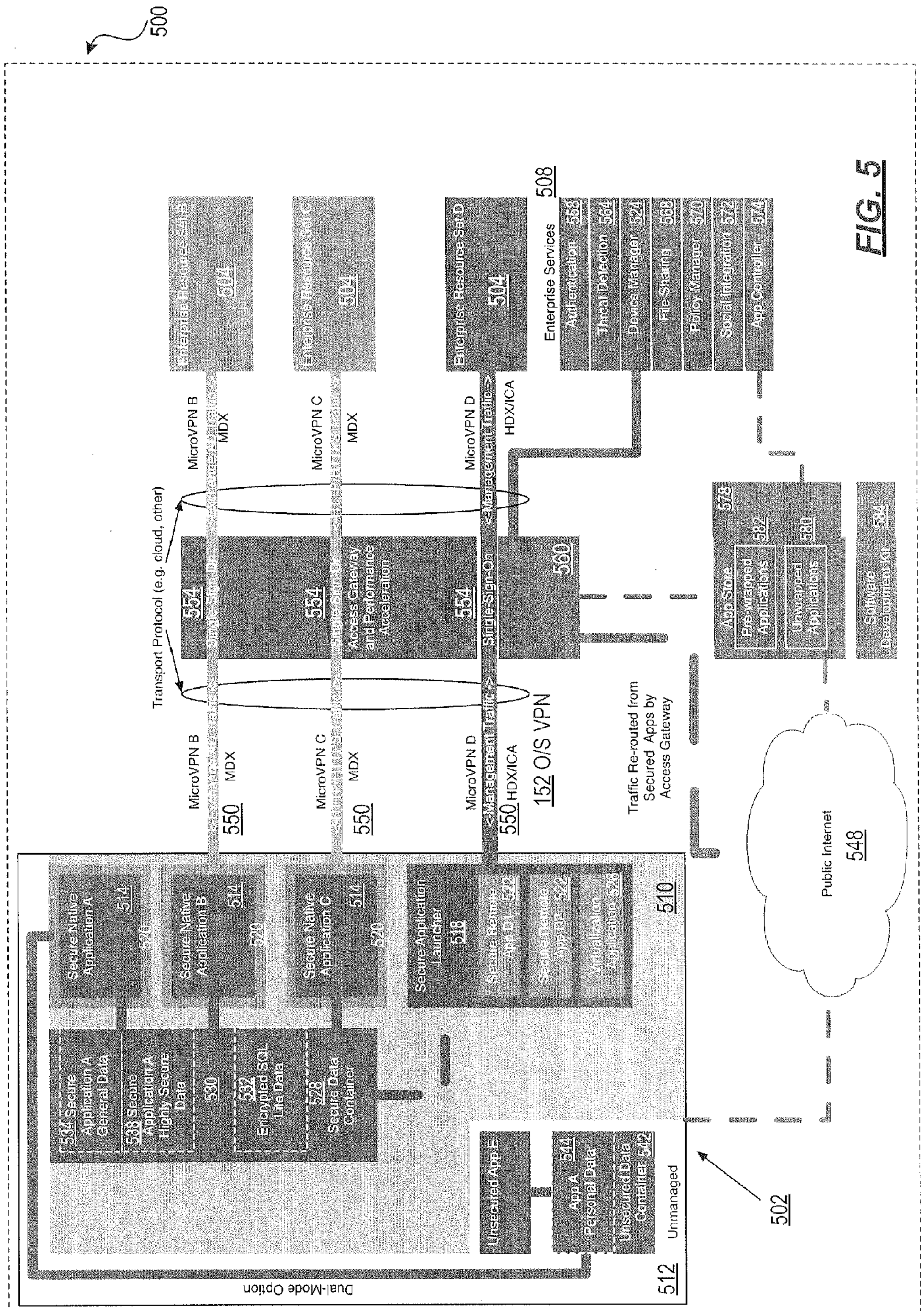

411
412
413
414
Management Server
410
401
403a
404a
405a
Storage A
Storage B
Storage C
Storage D
Network Element A
Network Element B
Network Element C
Network Element D
402
403b
404b
403b
Storage A
Storage B
Storage C
Storage D
Network Element A
Network Element B
Network Element C
Network Element D

FIG. 4



500
Enterprise Resource Set B
504
Enterprise Resource Set C
504
Enterprise Resource Set D
504
Enterprise Services
508
Authentication 558
Threat Detection 564
Device Manager 524
File Sharing 568
Policy Manager 570
Social Integration 572
App Controller 574
MicroVPN B
MDX
MicroVPN C
MDX
MicroVPN D
HDX/ICA
<Management Traffic>
Transport Protocol (e.g. cloud, other)
554
Single-Sign-On
Access Gateway and Performance Acceleration
560
App Store
578
Pre-wrapped Applications 582
Unwrapped Applications 580
Software Development Kit 584
Traffic Re-routed from Secured Apps by Access Gateway
Public Internet
548
550
152 O/S VPN
Secure Native Application A
Secure Native Application B
Secure Native Application C
514
520
Secure Application Launcher
518
Secure Remote App D1 522
Secure Remote App D2 522
Virtualization Application 526
510
534 Secure Application A General Data
538 Secure Application A Highly Secure Data
530
532 Encrypted SQL Lite Data
528 Secure Data Container
Unsecured App E
App A Personal Data 544
Unsecured Data Container 542
Unmanaged
502
512
Dual-Mode Option

FIG. 5

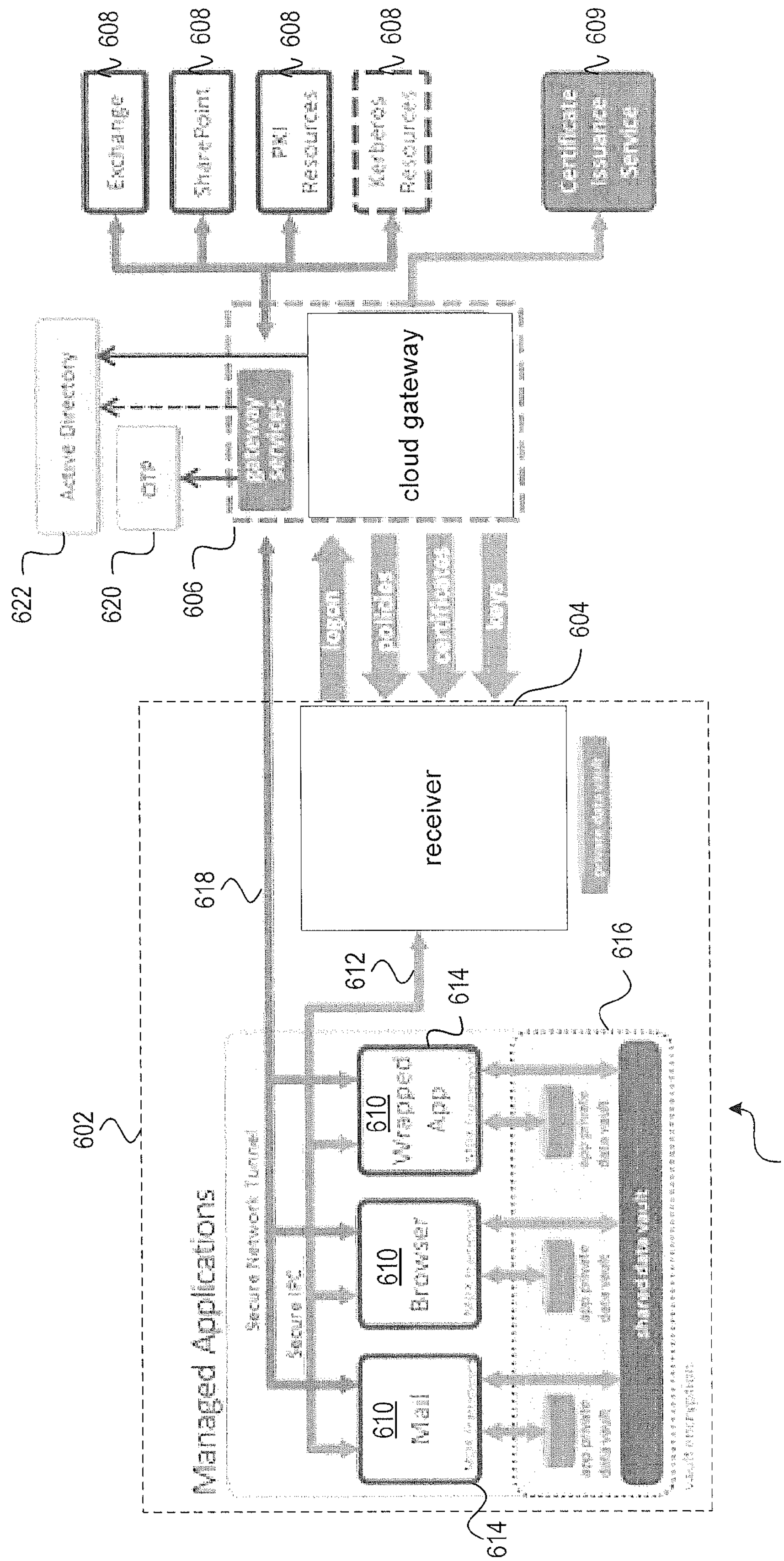

608
Exchange
SharePoint
PKI Resources
Kerberos Resources
609
Certificate Issuance Service
622
Active Directory
620
OTP
606
gateway services
cloud gateway
logon
policies
certificates
keys
604
receiver
602
618
612
614
616
Managed Applications
Secure Network Tunnel
Secure IPC
610 Mail
610 Browser
610 Wrapped App
shared data vault
600

FIG. 6

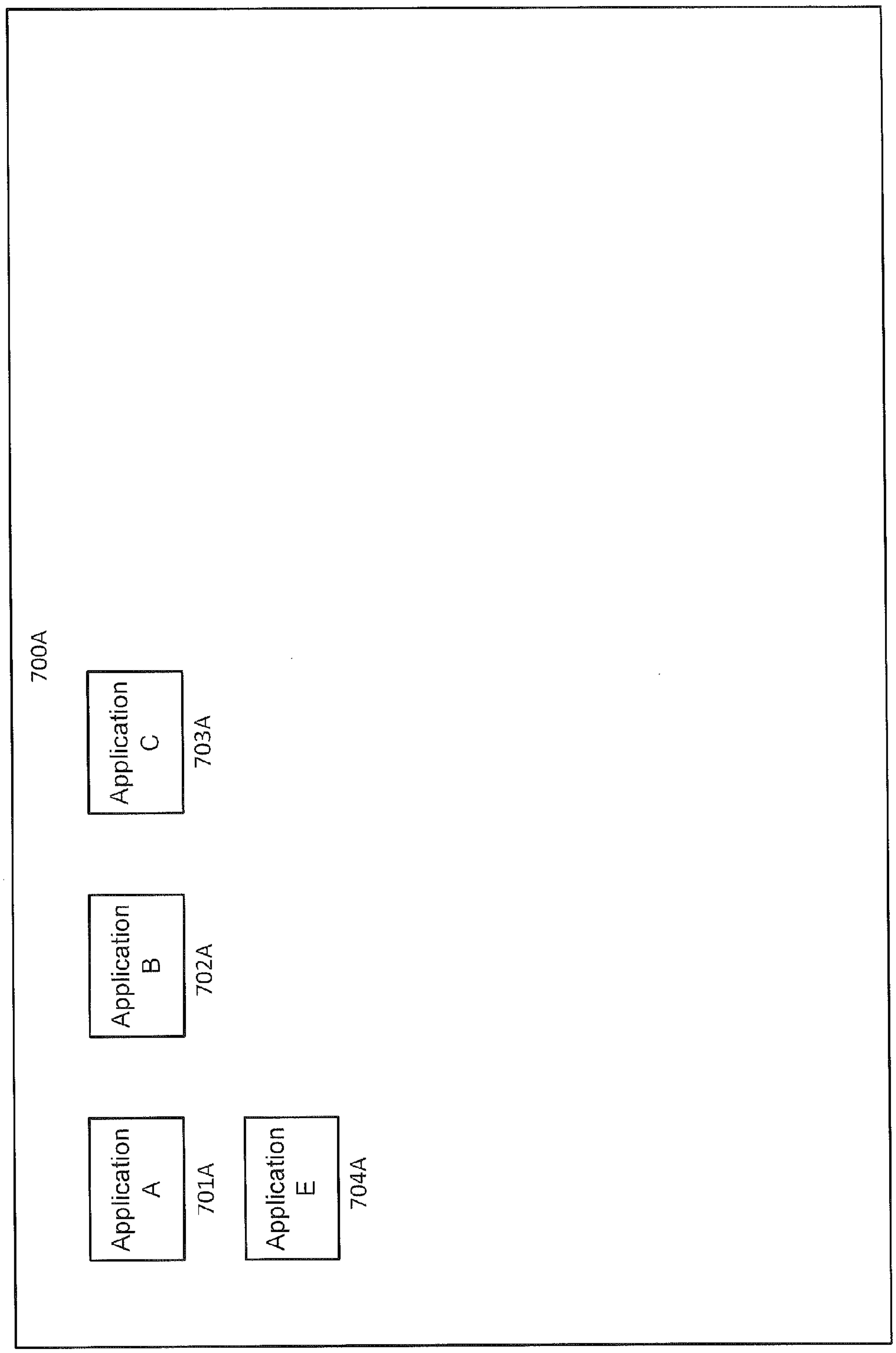

700A
Application C
703A
Application B
702A
Application A
701A
Application E
704A

FIG. 7A

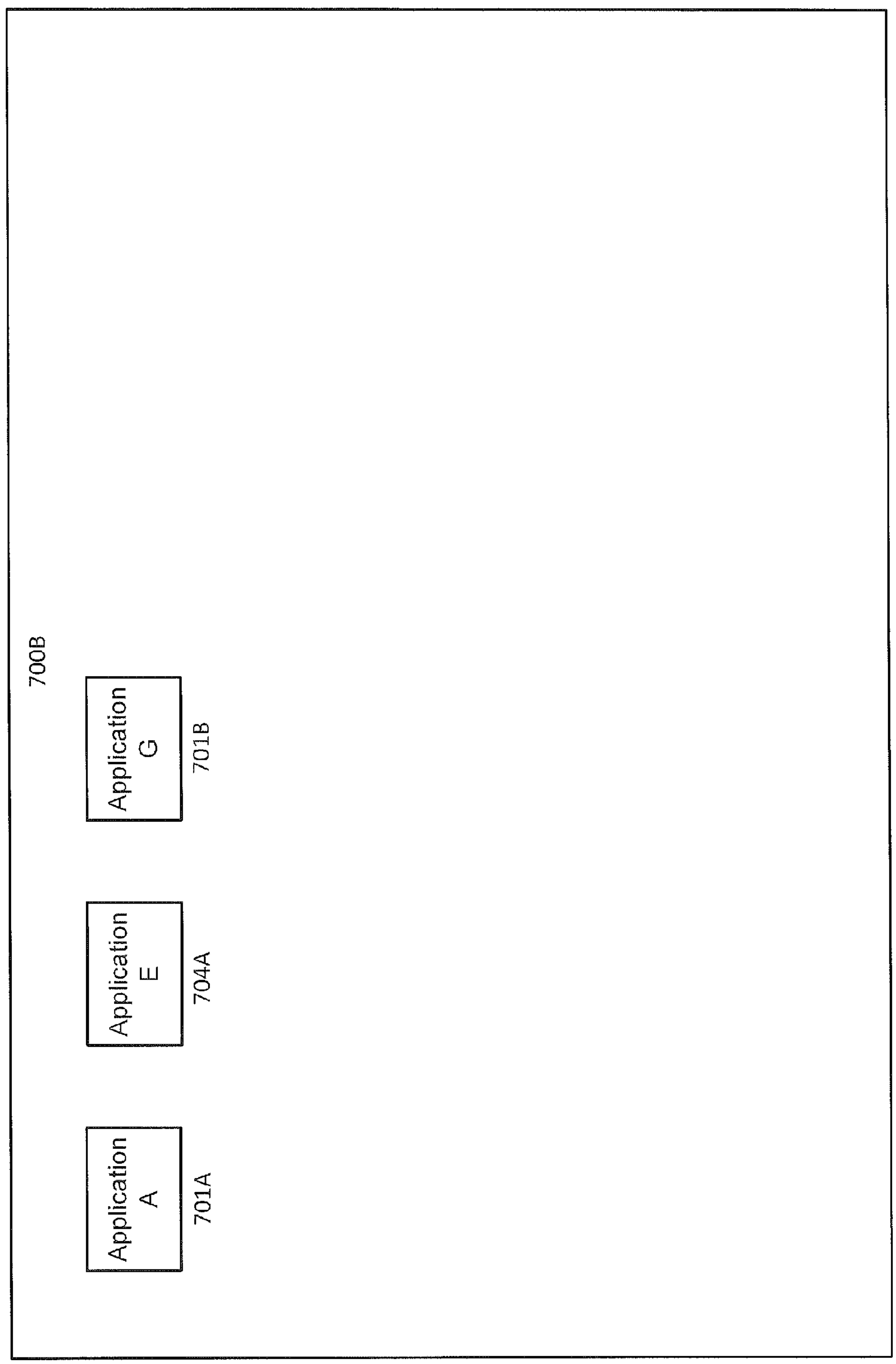

700B
Application G
701B
Application E
704A
Application A
701A

FIG. 7B

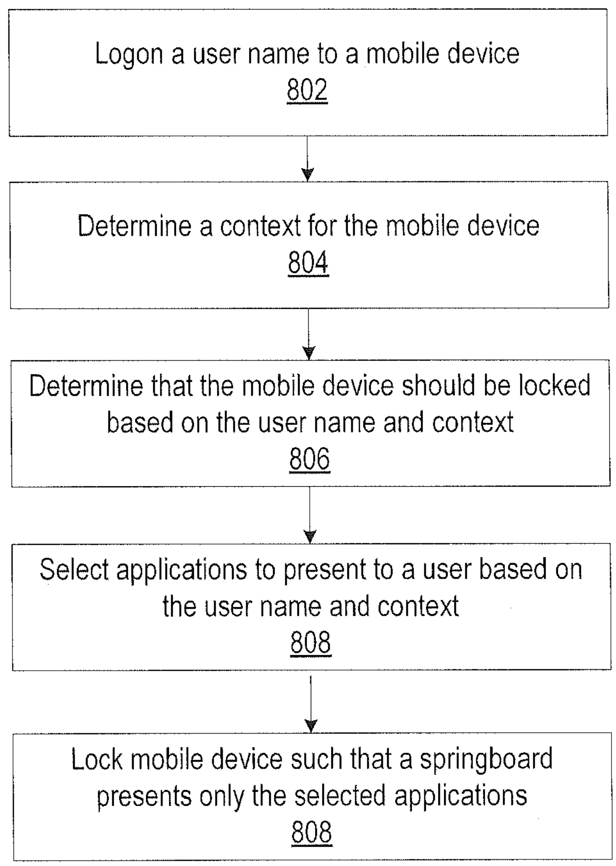

Logon a user name to a mobile device
802
Determine a context for the mobile device
804
Determine that the mobile device should be locked based on the user name and context
806
Select applications to present to a user based on the user name and context
808
Lock mobile device such that a springboard presents only the selected applications
808

FIG. 8

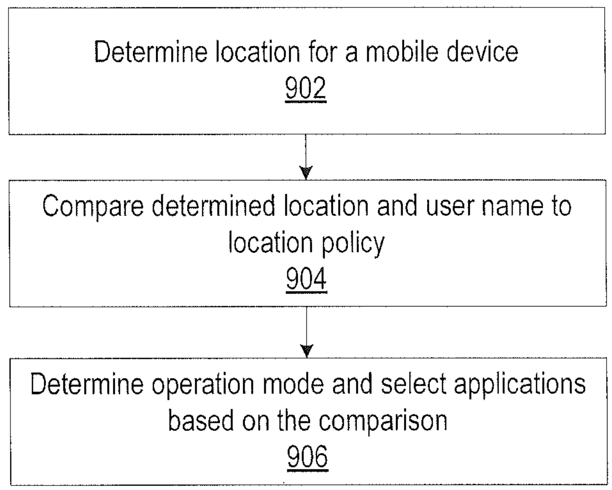

Determine location for a mobile device
902
Compare determined location and user name to location policy
904
Determine operation mode and select applications based on the comparison
906

FIG. 9

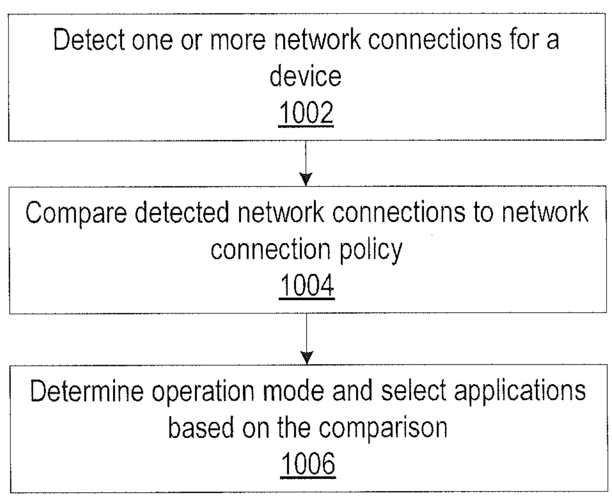

Detect one or more network connections for a device
1002
Compare detected network connections to network connection policy
1004
Determine operation mode and select applications based on the comparison
1006

FIG. 10

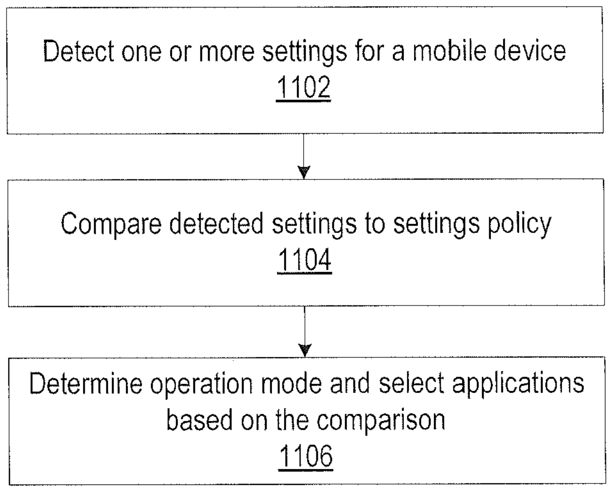

Detect one or more settings for a mobile device
1102
Compare detected settings to settings policy
1104
Determine operation mode and select applications based on the comparison
1106

FIG. 11

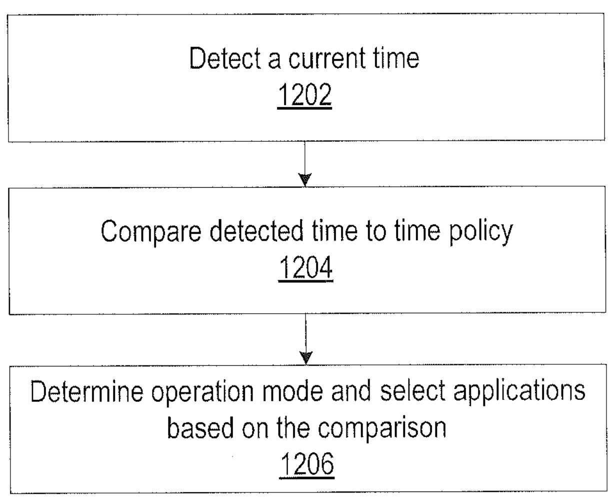

Detect a current time
1202
Compare detected time to time policy
1204
Determine operation mode and select applications based on the comparison
1206

FIG. 12

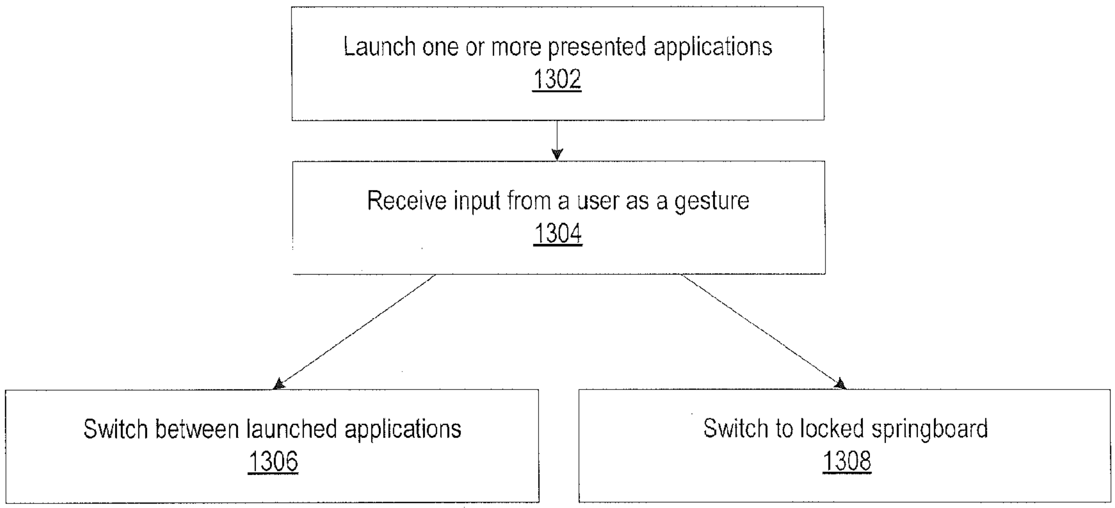

Launch one or more presented applications
1302
Receive input from a user as a gesture
1304
Switch between launched applications
1306
Switch to locked springboard
1308

FIG. 13

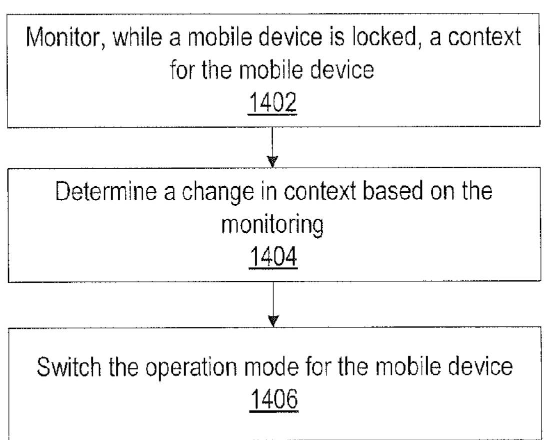

Monitor, while a mobile device is locked, a context for the mobile device
1402
Determine a change in context based on the monitoring
1404
Switch the operation mode for the mobile device
1406

FIG. 14

MOBILE DEVICE LOCKING WITH CONTEXT

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to provisional U.S. Provisional Application Ser. No. 61/806,557, filed Mar. 29, 2013, and entitled "Systems and Methods for Enterprise Mobility Management," which is herein incorporated by reference in its entirety.

BACKGROUND

The use of mobile computing devices continues to grow. In particular, business and other enterprises have come to rely on mobile computing devices to allow individuals to remotely access various enterprise resources. Such resources may include, for example, electronic mail services, file services, data, and other electronic resources provided by the computer systems of an enterprise.

With this insurgence of business use, individuals are beginning to use their mobile computing devices in both business and personal ways. For example, an employee of a corporation may access a corporate email account and a personal email account from the same mobile computing device.

SUMMARY

The following presents a simplified summary of various aspects described herein. This summary is not an extensive overview, and is not intended to identify key or critical elements or to delineate the scope of the claims. The following summary merely presents some concepts in a simplified form as an introductory prelude to the more detailed description provided below.

The ability to control content available on a mobile device in certain circumstances, for instance when a mobile device is located at a corporate premises, would be advantageous. Accordingly, methods and systems for locking a mobile device on an interface are described. A user logs on to a mobile device with a user name. The mobile device then determines a context for the mobile device based on one or more operational parameters of the device. For example, a context for the mobile device may be a current location of the device. Based on the context and user name, the mobile device may run in locked mode. In locked mode, applications are selected to be presented on the mobile device based on the user name and context. The mobile device is locked on a springboard that presents only the selected applications to the user for launching.

In an embodiment, the context for the mobile device may comprise a location for the mobile device that will be running the selected application, one or more network connections for the mobile device, one or more settings for the mobile device and a current time. One or more of these contexts combined with the user name may be compared to policies to determine whether the mobile device should be locked and which applications to present.

In another embodiment, an operation mode may be switched for a mobile device. One or more contexts may be monitored for the mobile device while the mobile device is locked and a change in operation mode may be detected based on the monitoring. For example, one or more contexts may change for the mobile device and a policy may define that an operation mode for the mobile device is to be changed to unlocked. Accordingly, the operation mode may be switched to unlocked.

These and additional aspects will be appreciated with the benefit of the disclosures discussed in further detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of aspects described herein and the advantages thereof may be acquired by referring to the following description in consideration of the accompanying drawings, in which like reference numbers indicate like features, and wherein:

FIG. **1** depicts an illustrative computer system architecture that may be used in accordance with an embodiment.

FIG. **2** depicts an illustrative remote-access system architecture that may be used in accordance with an embodiment.

FIG. **3** depicts an illustrative virtualized (hypervisor) system architecture that may be used in accordance with an embodiment.

FIG. **4** depicts an illustrative cloud-based system architecture that may be used in accordance with an embodiment.

FIG. **5** depicts an illustrative enterprise mobility management system that may be used in accordance with an embodiment.

FIG. **6** depicts another illustrative enterprise mobility management system that may be used in accordance with an embodiment.

FIGS. **7**A and **7**B depict sample interfaces of a mobile device in accordance with an embodiment.

FIG. **8** is a flowchart for locking a mobile device based on context and user name in accordance with an embodiment.

FIG. **9** is a flowchart for determining a location context for an application a mobile device in accordance with an embodiment.

FIG. **10** is a flowchart for determining a network connection context for a mobile device in accordance with an embodiment.

FIG. **11** is a flowchart for determining a settings context for a mobile device in accordance with an embodiment.

FIG. **12** is a flowchart for determining a current time context for a mobile device in accordance with an embodiment.

FIG. **13** is a flowchart for navigating a locked mobile device in accordance with an embodiment.

FIG. **14** is a flowchart for switching an operation mode for a mobile device in accordance with an embodiment.

DETAILED DESCRIPTION

In the following description of the various embodiments, reference is made to the accompanying drawings identified above and which form a part hereof, and in which is shown by way of illustration various embodiments in which aspects described herein may be practiced. It is to be understood that other embodiments may be utilized and structural and functional modifications may be made without departing from the scope described herein. Various aspects are capable of other embodiments and of being practiced or being carried out in various different ways.

As a general introduction to the subject matter described in more detail below, aspects described herein are directed towards controlling remote access to resources at an enterprise computing system using managed mobile applications at mobile computing devices. An access manager may perform a validation process that determines whether a mobile application requesting access to enterprise resources has accurately identified itself and has not been subsequently altered after installation at the mobile computing device. In this way, the access manager may ensure the mobile application requesting access to the enterprise resource can be trusted and is not attempting to circumvent the security mechanisms used to protect those enterprise resources. As a result, individuals associated with the enterprise may advantageously utilize enterprise resources at their personal mobile devices.

It is to be understood that the phraseology and terminology used herein are for the purpose of description and should not be regarded as limiting. Rather, the phrases and terms used herein are to be given their broadest interpretation and meaning. The use of "including" and "comprising" and variations thereof is meant to encompass the items listed thereafter and equivalents thereof as well as additional items and equivalents thereof. The use of the terms "mounted," "connected," "coupled," "positioned," "engaged" and similar terms, is meant to include both direct and indirect mounting, connecting, coupling, positioning and engaging.

Computing Architecture

Computer software, hardware, and networks may be utilized in a variety of different system environments, including standalone, networked, remote-access (aka, remote desktop), virtualized, and/or cloud-based environments, among others. FIG. **1** illustrates one example of a system architecture and data processing device that may be used to implement one or more illustrative aspects described herein in a standalone and/or networked environment. Various network nodes **103**, **105**, **107**, and **109** may be interconnected via a wide area network (WAN) **101**, such as the Internet. Other networks may also or alternatively be used, including private intranets, corporate networks, LANs, metropolitan area networks (MAN) wireless networks, personal networks (PAN), and the like. Network **101** is for illustration purposes and may be replaced with fewer or additional computer networks. A local area network (LAN) may have one or more of any known LAN topology and may use one or more of a variety of different protocols, such as Ethernet. Devices **103**, **105**, **107**, **109** and other devices (not shown) may be connected to one or more of the networks via twisted pair wires, coaxial cable, fiber optics, radio waves or other communication media.

The term "network" as used herein and depicted in the drawings refers not only to systems in which remote storage devices are coupled together via one or more communication paths, but also to stand-alone devices that may be coupled, from time to time, to such systems that have storage capability. Consequently, the term "network" includes not only a "physical network" but also a "content network," which is comprised of the data—attributable to a single entity—which resides across all physical networks.

The components may include data server **103**, web server **105**, and client computers **107**, **109**. Data server **103** provides overall access, control and administration of databases and control software for performing one or more illustrative aspects describe herein. Data server **103** may be connected to web server **105** through which users interact with and obtain data as requested. Alternatively, data server **103** may act as a web server itself and be directly connected to the Internet. Data server **103** may be connected to web server **105** through the network **101** (e.g., the Internet), via direct or indirect connection, or via some other network. Users may interact with the data server **103** using remote computers **107**, **109**, e.g., using a web browser to connect to the data server **103** via one or more externally exposed web sites hosted by web server **105**. Client computers **107**, **109** may be used in concert with data server **103** to access data stored therein, or may be used for other purposes. For example, from client device **107** a user may access web server **105** using an Internet browser, as is known in the art, or by executing a software application that communicates with web server **105** and/or data server **103** over a computer network (such as the Internet).

Servers and applications may be combined on the same physical machines, and retain separate virtual or logical addresses, or may reside on separate physical machines. FIG. **1** illustrates just one example of a network architecture that may be used, and those of skill in the art will appreciate that the specific network architecture and data processing devices used may vary, and are secondary to the functionality that they provide, as further described herein. For example, services provided by web server **105** and data server **103** may be combined on a single server.

Each component **103**, **105**, **107**, **109** may be any type of known computer, server, or data processing device. Data server **103**, e.g., may include a processor **111** controlling overall operation of the rate server **103**. Data server **103** may further include RAM **113**, ROM **115**, network interface **117**, input/output interfaces **119** (e.g., keyboard, mouse, display, printer, etc.), and memory **121**. I/O **119** may include a variety of interface units and drives for reading, writing, displaying, and/or printing data or files. Memory **121** may further store operating system software **123** for controlling overall operation of the data processing device **103**, control logic **125** for instructing data server **103** to perform aspects described herein, and other application software **127** providing secondary, support, and/or other functionality which may or may not be used in conjunction with aspects described herein. The control logic may also be referred to herein as the data server software **125**. Functionality of the data server software may refer to operations or decisions made automatically based on rules coded into the control logic, made manually by a user providing input into the system, and/or a combination of automatic processing based on user input (e.g., queries, data updates, etc.).

Memory **121** may also store data used in performance of one or more aspects described herein, including a first database **129** and a second database **131**. In some embodiments, the first database may include the second database (e.g., as a separate table, report, etc.). That is, the information can be stored in a single database, or separated into different logical, virtual, or physical databases, depending on system design. Devices **105**, **107**, **109** may have similar or different architecture as described with respect to device **103**. Those of skill in the art will appreciate that the functionality of data processing device **103** (or device **105**, **107**, **109**) as described herein may be spread across multiple data processing devices, for example, to distribute processing load across multiple computers, to segregate transactions based on geographic location, user access level, quality of service (QoS), etc.

One or more aspects may be embodied in computer-usable or readable data and/or computer-executable instructions, such as in one or more program modules, executed by one or more computers or other devices as described herein. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types when executed by a processor in a computer or other device. The modules may be written in a source code programming language that is subsequently compiled for execution, or may be written in a scripting language such as (but not limited to) HTML or XML. The computer executable instructions may be stored on a computer readable medium such as a nonvolatile storage device. Any suitable computer readable storage media may be utilized, including hard disks, CD-ROMs, optical storage devices, magnetic storage devices, and/or any combination thereof. In addition, various transmission (non-storage) media representing data or events as described herein may be transferred between a source and a destination in the form of electromagnetic waves traveling through signal-conducting media such as metal wires, optical fibers, and/or wireless transmission media (e.g., air and/or space). Various aspects described herein may be embodied as a method, a data processing system, or a computer program product. Therefore, various functionalities may be embodied in whole or in part in software, firmware and/or hardware or hardware equivalents such as integrated circuits, field programmable gate arrays (FPGA), and the like. Particular data structures may be used to more effectively implement one or more aspects described herein, and such data structures are contemplated within the scope of computer executable instructions and computer-usable data described herein.

With further reference to FIG. **2**, one or more aspects described herein may be implemented in a remote-access environment. FIG. **2** depicts an example system architecture including a generic computing device **201** in an illustrative computing environment **200** that may be used according to one or more illustrative aspects described herein. Generic computing device **201** may be used as a server **206***a* in a single-server or multi-server desktop virtualization system (e.g., a remote access or cloud system) configured to provide virtual machines for client access devices. The generic computing device **201** may have a processor **203** for controlling overall operation of the server and its associated components, including random access memory (RAM) **205**, read-only memory (ROM) **207**, input/output (I/O) module **209**, and memory **215**.

I/O module **209** may include a mouse, keypad, touch screen, scanner, optical reader, and/or stylus (or other input device(s)) through which a user of generic computing device **201** may provide input, and may also include one or more of a speaker for providing audio output and a video display device for providing textual, audiovisual, and/or graphical output. Software may be stored within memory **215** and/or other storage to provide instructions to processor **203** for configuring generic computing device **201** into a special purpose computing device in order to perform various functions as described herein. For example, memory **215** may store software used by the computing device **201**, such as an operating system **217**, application programs **219**, and an associated database **221**.

Computing device **201** may operate in a networked environment supporting connections to one or more remote computers, such as terminals **240** (also referred to as client devices). The terminals **240** may be personal computers, mobile devices, laptop computers, tablets, or servers that include many or all of the elements described above with respect to the generic computing device **103** or **201**. The network connections depicted in FIG. **2** include a local area network (LAN) **225** and a wide area network (WAN) **229**, but may also include other networks. When used in a LAN networking environment, computing device **201** may be connected to the LAN **225** through a network interface or adapter **223**. When used in a WAN networking environment, computing device **201** may include a modem **227** or other wide area network interface for establishing communications over the WAN **229**, such as computer network **230** (e.g., the Internet). It will be appreciated that the network connections shown are illustrative and other means of establishing a communications link between the computers may be used. Computing device **201** and/or terminals **240** may also be mobile terminals (e.g., mobile phones, smartphones, PDAs, notebooks, etc.) including various other components, such as a battery, speaker, and antennas (not shown).

Aspects described herein may also be operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of other computing systems, environments, and/or configurations that may be suitable for use with aspects described herein include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

As shown in FIG. **2**, one or more client devices **240** may be in communication with one or more servers **206***a*-**206***n* (generally referred to herein as "server(s) **206**"). In one embodiment, the computing environment **200** may include a network appliance installed between the server(s) **206** and client machine(s) **240**. The network appliance may manage client/server connections, and in some cases can load balance client connections amongst a plurality of backend servers **206**.

The client machine(s) **240** may in some embodiments be referred to as a single client machine **240** or a single group of client machines **240**, while server(s) **206** may be referred to as a single server **206** or a single group of servers **206**. In one embodiment a single client machine **240** communicates with more than one server **206**, while in another embodiment a single server **206** communicates with more than one client machine **240**. In yet another embodiment, a single client machine **240** communicates with a single server **206**.

A client machine **240** can, in some embodiments, be referenced by any one of the following non-exhaustive terms: client machine(s); client(s); client computer(s); client device(s); client computing device(s); local machine; remote machine; client node(s); endpoint(s); or endpoint node(s). The server **206**, in some embodiments, may be referenced by any one of the following non-exhaustive terms: server(s), local machine; remote machine; server farm(s), or host computing device(s).

In one embodiment, the client machine **240** may be a virtual machine. The virtual machine may be any virtual machine, while in some embodiments the virtual machine may be any virtual machine managed by a Type 1 or Type 2 hypervisor, for example, a hypervisor developed by Citrix Systems, IBM, VMware, or any other hypervisor. In some aspects, the virtual machine may be managed by a hypervisor, while in aspects the virtual machine may be managed by a hypervisor executing on a server **206** or a hypervisor executing on a client **240**.

Some embodiments include a client device **240** that displays application output generated by an application remotely executing on a server **206** or other remotely located machine. In these embodiments, the client device **240** may execute a virtual machine receiver program or application to display the output in an application window, a browser, or other output window. In one example, the application is a desktop, while in other examples the application is an application that generates or presents a desktop. A desktop may include a graphical shell providing a user interface for an instance of an operating system in which local and/or remote applications can be integrated. Applications, as used herein, are programs that execute after an instance of an operating system (and, optionally, also the desktop) has been loaded.

The server **206**, in some embodiments, uses a remote presentation protocol or other program to send data to a thin-client or remote-display application executing on the client to present display output generated by an application executing on the server **206**. The thin-client or remote-display protocol can be any one of the following non-exhaustive list of protocols: the Independent Computing Architecture (ICA) protocol developed by Citrix Systems, Inc. of Ft. Lauderdale, Fla.; or the Remote Desktop Protocol (RDP) manufactured by the Microsoft Corporation of Redmond, Wash.

A remote computing environment may include more than one server **206***a*-**206***n* such that the servers **206***a*-**206***n* are logically grouped together into a server farm **206**, for example, in a cloud computing environment. The server farm **206** may include servers **206** that are geographically dispersed while and logically grouped together, or servers **206** that are located proximate to each other while logically grouped together. Geographically dispersed servers **206***a*-**206***n* within a server farm **206** can, in some embodiments, communicate using a WAN (wide), MAN (metropolitan), or LAN (local), where different geographic regions can be characterized as: different continents; different regions of a continent; different countries; different states; different cities; different campuses; different rooms; or any combination of the preceding geographical locations. In some embodiments the server farm **206** may be administered as a single entity, while in other embodiments the server farm **206** can include multiple server farms.

In some embodiments, a server farm may include servers **206** that execute a substantially similar type of operating system platform (e.g., WINDOWS, UNIX, LINUX, iOS, ANDROID, SYMBIAN, etc.) In other embodiments, server farm **206** may include a first group of one or more servers that execute a first type of operating system platform, and a second group of one or more servers that execute a second type of operating system platform.

Server **206** may be configured as any type of server, as needed, e.g., a file server, an application server, a web server, a proxy server, an appliance, a network appliance, a gateway, an application gateway, a gateway server, a virtualization server, a deployment server, a SSL VPN server, a firewall, a web server, an application server or as a master application server, a server executing an active directory, or a server executing an application acceleration program that provides firewall functionality, application functionality, or load balancing functionality. Other server types may also be used.

Some embodiments include a first server **106***a* that receives requests from a client machine **240**, forwards the request to a second server **106***b*, and responds to the request generated by the client machine **240** with a response from the second server **106***b*. First server **106***a* may acquire an enumeration of applications available to the client machine **240** and well as address information associated with an application server **206** hosting an application identified within the enumeration of applications. First server **106***a* can then present a response to the client's request using a web interface, and communicate directly with the client **240** to provide the client **240** with access to an identified application. One or more clients **240** and/or one or more servers **206** may transmit data over network **230**, e.g., network **101**.

FIG. **2** shows a high-level architecture of an illustrative desktop virtualization system. As shown, the desktop virtualization system may be single-server or multi-server system, or cloud system, including at least one virtualization server **206** configured to provide virtual desktops and/or virtual applications to one or more client access devices **240**. As used herein, a desktop refers to a graphical environment or space in which one or more applications may be hosted and/or executed. A desktop may include a graphical shell providing a user interface for an instance of an operating system in which local and/or remote applications can be integrated. Applications may include programs that execute after an instance of an operating system (and, optionally, also the desktop) has been loaded. Each instance of the operating system may be physical (e.g., one operating system per device) or virtual (e.g., many instances of an OS running on a single device). Each application may be executed on a local device, or executed on a remotely located device (e.g., remoted).

With further reference to FIG. **3**, a computer device **301** may be configured as a virtualization server in a virtualization environment, for example, a single-server, multi-server, or cloud computing environment. Virtualization server **301** illustrated in FIG. **3** can be deployed as and/or implemented by one or more embodiments of the server **206** illustrated in FIG. **2** or by other known computing devices. Included in virtualization server **301** is a hardware layer that can include one or more physical disks **304**, one or more physical devices **306**, one or more physical processors **308** and one or more physical memories **316**. In some embodiments, firmware **312** can be stored within a memory element in the physical memory **316** and can be executed by one or more of the physical processors **308**. Virtualization server **301** may further include an operating system **314** that may be stored in a memory element in the physical memory **316** and executed by one or more of the physical processors **308**. Still further, a hypervisor **302** may be stored in a memory element in the physical memory **316** and can be executed by one or more of the physical processors **308**.

Executing on one or more of the physical processors **308** may be one or more virtual machines **332**A-C (generally **332**). Each virtual machine **332** may have a virtual disk **326**A-C and a virtual processor **328**A-C. In some embodiments, a first virtual machine **332**A may execute, using a virtual processor **328**A, a control program **320** that includes a tools stack **324**. Control program **320** may be referred to as a control virtual machine, Dom0, Domain 0, or other virtual machine used for system administration and/or control. In some embodiments, one or more virtual machines **332**B-C can execute, using a virtual processor **328**B-C, a guest operating system **330**A-B.

Virtualization server **301** may include a hardware layer **310** with one or more pieces of hardware that communicate with the virtualization server **301**. In some embodiments, the hardware layer **310** can include one or more physical disks **304**, one or more physical devices **306**, one or more physical processors **308**, and one or more memory **216**. Physical components **304**, **306**, **308**, and **316** may include, for example, any of the components described above. Physical devices **306** may include, for example, a network interface card, a video card, a keyboard, a mouse, an input device, a monitor, a display device, speakers, an optical drive, a storage device, a universal serial bus connection, a printer, a scanner, a network element (e.g., router, firewall, network address translator, load balancer, virtual private network (VPN) gateway, Dynamic Host Configuration Protocol (DHCP) router, etc.), or any device connected to or communicating with virtualization server **301**. Physical memory **316** in the hardware layer **310** may include any type of memory. Physical memory **316** may store data, and in some embodiments may store one or more programs, or set of executable instructions. FIG. **3** illustrates an embodiment where firmware **312** is stored within the physical memory **316** of virtualization server **301**. Programs or executable instructions stored in the physical memory **316** can be executed by the one or more processors **308** of virtualization server **301**.

Virtualization server **301** may also include a hypervisor **302**. In some embodiments, hypervisor **302** may be a program executed by processors **308** on virtualization server **301** to create and manage any number of virtual machines **332**. Hypervisor **302** may be referred to as a virtual machine monitor, or platform virtualization software. In some embodiments, hypervisor **302** can be any combination of executable instructions and hardware that monitors virtual machines executing on a computing machine. Hypervisor **302** may be Type 2 hypervisor, where the hypervisor that executes within an operating system **314** executing on the virtualization server **301**. Virtual machines then execute at a level above the hypervisor. In some embodiments, the Type 2 hypervisor executes within the context of a user's operating system such that the Type 2 hypervisor interacts with the user's operating system. In other embodiments, one or more virtualization servers **201** in a virtualization environment may instead include a Type 1 hypervisor (Not Shown). A Type 1 hypervisor may execute on the virtualization server **301** by directly accessing the hardware and resources within the hardware layer **310**. That is, while a Type 2 hypervisor **302** accesses system resources through a host operating system **314**, as shown, a Type 1 hypervisor may directly access all system resources without the host operating system **314**. A Type 1 hypervisor may execute directly on one or more physical processors **308** of virtualization server **301**, and may include program data stored in the physical memory **316**.

Hypervisor **302**, in some embodiments, can provide virtual resources to operating systems **330** or control programs **320** executing on virtual machines **332** in any manner that simulates the operating systems **330** or control programs **320** having direct access to system resources. System resources can include, but are not limited to, physical devices **306**, physical disks **304**, physical processors **308**, physical memory **316** and any other component included in virtualization server **301** hardware layer **310**. Hypervisor **302** may be used to emulate virtual hardware, partition physical hardware, virtualize physical hardware, and/or execute virtual machines that provide access to computing environments. In still other embodiments, hypervisor **302** controls processor scheduling and memory partitioning for a virtual machine **332** executing on virtualization server **301**. Hypervisor **302** may include those manufactured by VMWare, Inc., of Palo Alto, Calif.; the XEN hypervisor, an open source product whose development is overseen by the open source Xen.org community; HyperV, VirtualServer or virtual PC hypervisors provided by Microsoft, or others. In some embodiments, virtualization server **301** executes a hypervisor **302** that creates a virtual machine platform on which guest operating systems may execute. In these embodiments, the virtualization server **301** may be referred to as a host server. An example of such a virtualization server is the XEN SERVER provided by Citrix Systems, Inc., of Fort Lauderdale, Fla.

Hypervisor **302** may create one or more virtual machines **332**B-C (generally **332**) in which guest operating systems **330** execute. In some embodiments, hypervisor **302** may load a virtual machine image to create a virtual machine **332**. In other embodiments, the hypervisor **302** may executes a guest operating system **330** within virtual machine **332**. In still other embodiments, virtual machine **332** may execute guest operating system **330**.

In addition to creating virtual machines **332**, hypervisor **302** may control the execution of at least one virtual machine **332**. In other embodiments, hypervisor **302** may presents at least one virtual machine **332** with an abstraction of at least one hardware resource provided by the virtualization server **301** (e.g., any hardware resource available within the hardware layer **310**). In other embodiments, hypervisor **302** may control the manner in which virtual machines **332** access physical processors **308** available in virtualization server **301**. Controlling access to physical processors **308** may include determining whether a virtual machine **332** should have access to a processor **308**, and how physical processor capabilities are presented to the virtual machine **332**.

As shown in FIG. **3**, virtualization server **301** may host or execute one or more virtual machines **332**. A virtual machine **332** is a set of executable instructions that, when executed by a processor **308**, imitate the operation of a physical computer such that the virtual machine **332** can execute programs and processes much like a physical computing device. While FIG. **3** illustrates an embodiment where a virtualization server **301** hosts three virtual machines **332**, in other embodiments virtualization server **301** can host any number of virtual machines **332**. Hypervisor **302**, in some embodiments, provides each virtual machine **332** with a unique virtual view of the physical hardware, memory, processor and other system resources available to that virtual machine **332**. In some embodiments, the unique virtual view can be based on one or more of virtual machine permissions, application of a policy engine to one or more virtual machine identifiers, a user accessing a virtual machine, the applications executing on a virtual machine, networks accessed by a virtual machine, or any other desired criteria. For instance, hypervisor **302** may create one or more unsecure virtual machines **332** and one or more secure virtual machines **332**. Unsecure virtual machines **332** may be prevented from accessing resources, hardware, memory locations, and programs that secure virtual machines **332** may be permitted to access. In other embodiments, hypervisor **302** may provide each virtual machine **332** with a substantially similar virtual view of the physical hardware, memory, processor and other system resources available to the virtual machines **332**.

Each virtual machine **332** may include a virtual disk **326**A-C (generally **326**) and a virtual processor **328**A-C (generally **328**.) The virtual disk **326**, in some embodiments, is a virtualized view of one or more physical disks **304** of the virtualization server **301**, or a portion of one or more physical disks **304** of the virtualization server **301**. The virtualized view of the physical disks **304** can be generated, provided and managed by the hypervisor **302**. In some embodiments, hypervisor **302** provides each virtual machine **332** with a unique view of the physical disks **304**. Thus, in these embodiments, the particular virtual disk **326** included in each virtual machine **332** can be unique when compared with the other virtual disks **326**.

A virtual processor **328** can be a virtualized view of one or more physical processors **308** of the virtualization server **301**. In some embodiments, the virtualized view of the physical processors **308** can be generated, provided and managed by hypervisor **302**. In some embodiments, virtual processor **328** has substantially all of the same characteristics of at least one physical processor **308**. In other embodiments, virtual processor **308** provides a modified view of physical processors **308** such that at least some of the characteristics of the virtual processor **328** are different than the characteristics of the corresponding physical processor **308**.

With further reference to FIG. **4**, some aspects described herein may be implemented in a cloud-based environment. FIG. **4** illustrates an example of a cloud computing environment (or cloud system) **400**. As seen in FIG. **4**, client computers **411**-**414** may communicate with a cloud management server **410** to access the computing resources (e.g., host servers **403**, storage resources **404**, and network resources **405**) of the cloud system.

Management server **410** may be implemented on one or more physical servers. The management server **410** may run, for example, CLOUDSTACK by Citrix Systems, Inc. of Ft. Lauderdale, Fla., or OPENSTACK, among others. Management server **410** may manage various computing resources, including cloud hardware and software resources, for example, host computers **403**, data storage devices **404**, and networking devices **405**. The cloud hardware and software resources may include private and/or public components. For example, a cloud may be configured as a private cloud to be used by one or more particular customers or client computers **411-414** and/or over a private network. In other embodiments, public clouds or hybrid public-private clouds may be used by other customers over an open or hybrid networks.

Management server **410** may be configured to provide user interfaces through which cloud operators and cloud customers may interact with the cloud system. For example, the management server **410** may provide a set of APIs and/or one or more cloud operator console applications (e.g., web-based on standalone applications) with user interfaces to allow cloud operators to manage the cloud resources, configure the virtualization layer, manage customer accounts, and perform other cloud administration tasks. The management server **410** also may include a set of APIs and/or one or more customer console applications with user interfaces configured to receive cloud computing requests from end users via client computers **411-414**, for example, requests to create, modify, or destroy virtual machines within the cloud. Client computers **411-414** may connect to management server **410** via the Internet or other communication network, and may request access to one or more of the computing resources managed by management server **410**. In response to client requests, the management server **410** may include a resource manager configured to select and provision physical resources in the hardware layer of the cloud system based on the client requests. For example, the management server **410** and additional components of the cloud system may be configured to provision, create, and manage virtual machines and their operating environments (e.g., hypervisors, storage resources, services offered by the network elements, etc.) for customers at client computers **411-414**, over a network (e.g., the Internet), providing customers with computational resources, data storage services, networking capabilities, and computer platform and application support. Cloud systems also may be configured to provide various specific services, including security systems, development environments, user interfaces, and the like.

Certain clients **411-414** may be related, for example, different client computers creating virtual machines on behalf of the same end user, or different users affiliated with the same company or organization. In other examples, certain clients **411-414** may be unrelated, such as users affiliated with different companies or organizations. For unrelated clients, information on the virtual machines or storage of any one user may be hidden from other users.

Referring now to the physical hardware layer of a cloud computing environment, availability zones **401-402** (or zones) may refer to a collocated set of physical computing resources. Zones may be geographically separated from other zones in the overall cloud of computing resources. For example, zone **401** may be a first cloud datacenter located in California, and zone **402** may be a second cloud datacenter located in Florida. Management sever **410** may be located at one of the availability zones, or at a separate location. Each zone may include an internal network that interfaces with devices that are outside of the zone, such as the management server **410**, through a gateway. End users of the cloud (e.g., clients **411-414**) might or might not be aware of the distinctions between zones. For example, an end user may request the creation of a virtual machine having a specified amount of memory, processing power, and network capabilities. The management server **410** may respond to the user's request and may allocate the resources to create the virtual machine without the user knowing whether the virtual machine was created using resources from zone **401** or zone **402**. In other examples, the cloud system may allow end users to request that virtual machines (or other cloud resources) are allocated in a specific zone or on specific resources **403-405** within a zone.

In this example, each zone **401-402** may include an arrangement of various physical hardware components (or computing resources) **403-405**, for example, physical hosting resources (or processing resources), physical network resources, physical storage resources, switches, and additional hardware resources that may be used to provide cloud computing services to customers. The physical hosting resources in a cloud zone **401-402** may include one or more computer servers **403**, such as the virtualization servers **301** described above, which may be configured to create and host virtual machine instances. The physical network resources in a cloud zone **401** or **402** may include one or more network elements **405** (e.g., network service providers) comprising hardware and/or software configured to provide a network service to cloud customers, such as firewalls, network address translators, load balancers, virtual private network (VPN) gateways, Dynamic Host Configuration Protocol (DHCP) routers, and the like. The storage resources in the cloud zone **401-402** may include storage disks (e.g., solid state drives (SSDs), magnetic hard disks, etc.) and other storage devices.

The example cloud computing environment shown in FIG. **4** also may include a virtualization layer (e.g., as shown in FIGS. **1-3**) with additional hardware and/or software resources configured to create and manage virtual machines and provide other services to customers using the physical resources in the cloud. The virtualization layer may include hypervisors, as described above in FIG. **3**, along with other components to provide network virtualizations, storage virtualizations, etc. The virtualization layer may be as a separate layer from the physical resource layer, or may share some or all of the same hardware and/or software resources with the physical resource layer. For example, the virtualization layer may include a hypervisor installed in each of the virtualization servers **403** with the physical computing resources. Known cloud systems may alternatively be used, e.g., WINDOWS AZURE (Microsoft Corporation of Redmond Wash.), AMAZON EC2 (Amazon.com Inc. of Seattle, Wash.), IBM BLUE CLOUD (IBM Corporation of Armonk, N.Y.), or others.

Enterprise Mobility Management Architecture

FIG. **5** represents an enterprise mobility technical architecture **500** for use in a BYOD environment. The architecture enables a user of a mobile device **502** to both access enterprise or personal resources from a mobile device **502** and use the mobile device **502** for personal use. The user may access such enterprise resources **504** or enterprise services **508** using a mobile device **502** that is purchased by the user or a mobile device **502** that is provided by the enterprise to user. The user may utilize the mobile device **502** for business use only or for business and personal use. The mobile device may run an iOS operating system, and Android operating system, or the like. The enterprise may choose to implement policies to manage the mobile device **504**. The policies may be implanted through a firewall or gateway in such a way that the mobile device may be identified, secured or security verified, and provided selective or full access to the enterprise resources. The policies may be mobile device management policies, mobile application management policies, mobile data management policies, or some combination of mobile device, application, and data management policies. A mobile device **504** that is managed through the application of mobile device management policies may be referred to as an enrolled device.

The operating system of the mobile device may be separated into a managed partition **510** and an unmanaged partition **512**. The managed partition **510** may have policies applied to it to secure the applications running on and data stored in the managed partition. The applications running on the managed partition may be secure applications. The secure applications may be email applications, web browsing applications, software-as-a-service (SaaS) access applications, Windows Application access applications, and the like. The secure applications may be secure native applications **514**, secure remote applications **522** executed by a secure application launcher **518**, virtualization applications **526** executed by a secure application launcher **518**, and the like. The secure native applications **514** may be wrapped by a secure application wrapper **520**. The secure application wrapper **520** may include integrated policies that are executed on the mobile device **502** when the secure native application is executed on the device. The secure application wrapper **520** may include meta-data that points the secure native application **514** running on the mobile device **502** to the resources hosted at the enterprise that the secure native application **514** may require to complete the task requested upon execution of the secure native application **514**. The secure remote applications **522** executed by a secure application launcher **518** may be executed within the secure application launcher application **518**. The virtualization applications **526** executed by a secure application launcher **518** may utilize resources on the mobile device **502**, at the enterprise resources **504**, and the like. The resources used on the mobile device **502** by the virtualization applications **526** executed by a secure application launcher **518** may include user interaction resources, processing resources, and the like. The user interaction resources may be used to collect and transmit keyboard input, mouse input, camera input, tactile input, audio input, visual input, gesture input, and the like. The processing resources may be used to present a user interface, process data received from the enterprise resources **504**, and the like. The resources used at the enterprise resources **504** by the virtualization applications **526** executed by a secure application launcher **518** may include user interface generation resources, processing resources, and the like. The user interface generation resources may be used to assemble a user interface, modify a user interface, refresh a user interface, and the like. The processing resources may be used to create information, read information, update information, delete information, and the like. For example, the virtualization application may record user interactions associated with a GUI and communicate them to a server application where the server application will use the user interaction data as an input to the application operating on the server. In this arrangement, an enterprise may elect to maintain the application on the server side as well as data, files, etc. associated with the application. While an enterprise may elect to "mobilize" some applications in accordance with the principles herein by securing them for deployment on the mobile device, this arrangement may also be elected for certain applications. For example, while some applications may be secured for use on the mobile device, others may not be prepared or appropriate for deployment on the mobile device so the enterprise may elect to provide the mobile user access to the unprepared applications through virtualization techniques. As another example, the enterprise may have large complex applications with large and complex data sets (e.g. material resource planning applications) where it would be very difficult, or otherwise undesirable, to customize the application for the mobile device so the enterprise may elect to provide access to the application through virtualization techniques. As yet another example, the enterprise may have an application that maintains highly secured data (e.g. human resources data, customer data, engineering data) that may be deemed by the enterprise as too sensitive for even the secured mobile environment so the enterprise may elect to use virtualization techniques to permit mobile access to such applications and data. An enterprise may elect to provide both fully secured and fully functional applications on the mobile device as well as a virtualization application to allow access to applications that are deemed more properly operated on the server side. In an embodiment, the virtualization application may store some data, files, etc. on the mobile phone in one of the secure storage locations. An enterprise, for example, may elect to allow certain information to be stored on the phone while not permitting other information.

In connection with the virtualization application, as described herein, the mobile device may have a virtualization application that is designed to present GUI's and then record user interactions with the GUI. The application may communicate the user interactions to the server side to be used by the server side application as user interactions with the application. In response, the application on the server side may transmit back to the mobile device a new GUI. For example, the new GUI may be a static page, a dynamic page, an animation, or the like.

The applications running on the managed partition may be stabilized applications. The stabilized applications may be managed by a device manager **524**. The device manager **524** may monitor the stabilized applications and utilize techniques for detecting and remedying problems that would result in a destabilized application if such techniques were not utilized to detect and remedy the problems.

The secure applications may access data stored in a secure data container **528** in the managed partition **510** of the mobile device. The data secured in the secure data container may be accessed by the secure wrapped applications **514**, applications executed by a secure application launcher **522**, virtualization applications **526** executed by a secure application launcher **522**, and the like. The data stored in the secure data container **528** may include files, databases, and the like. The data stored in the secure data container **528** may include data restricted to a specific secure application **530**, shared among secure applications **532**, and the like. Data restricted to a secure application may include secure general data **534** and highly secure data **538**. Secure general data may use a strong form of encryption such as AES 128-bit encryption or the like, while highly secure data **538** may use a very strong form of encryption such as AES 254-bit encryption. Data stored in the secure data container **528** may be deleted from the device upon receipt of a command from the device manager **524**. The secure applications may have a dual-mode option **540**. The dual mode option **540** may present the user with an option to operate the secured application in an unsecured mode. In an unsecured mode, the secure applications may access data stored in an unsecured data container **542** on the unmanaged partition **512** of the mobile device **502**. The data stored in an unsecured data container may be personal data **544**. The data stored in an unsecured data container **542** may also be accessed by unsecured applications **548** that are running on the unmanaged partition **512** of the mobile device **502**. The data stored in an unsecured data container **542** may remain on the mobile device **502** when the data stored in the secure data container **528** is deleted from the mobile device **502**. An enterprise may want to delete from the mobile device selected or all data, files, and/or applications owned, licensed or controlled by the enterprise (enterprise data) while leaving or otherwise preserving personal data, files, and/or applications owned, licensed or controlled by the user (personal data). This operation may be referred to as a selective wipe. With the enterprise and personal data arranged in accordance to the aspects described herein, an enterprise may perform a selective wipe.

The mobile device may connect to enterprise resources **504** and enterprise services **508** at an enterprise, to the public Internet **548**, and the like. The mobile device may connect to enterprise resources **504** and enterprise services **508** through virtual private network connections. The virtual private network connections may be specific to particular applications **550**, particular devices, particular secured areas on the mobile device, and the like **552**. For example, each of the wrapped applications in the secured area of the phone may access enterprise resources through an application specific VPN such that access to the VPN would be granted based on attributes associated with the application, possibly in conjunction with user or device attribute information. The virtual private network connections may carry Microsoft Exchange traffic, Microsoft Active Directory traffic, HTTP traffic, HTTPS traffic, application management traffic, and the like. The virtual private network connections may support and enable single-sign-on authentication processes **554**. The single-sign-on processes may allow a user to provide a single set of authentication credentials, which are then verified by an authentication service **558**. The authentication service **558** may then grant to the user access to multiple enterprise resources **504**, without requiring the user to provide authentication credentials to each individual enterprise resource **504**.

The virtual private network connections may be established and managed by an access gateway **560**. The access gateway **560** may include performance enhancement features that manage, accelerate, and improve the delivery of enterprise resources **504** to the mobile device **502**. The access gateway may also re-route traffic from the mobile device **502** to the public Internet **548**, enabling the mobile device **502** to access publicly available and unsecured applications that run on the public Internet **548**. The mobile device may connect to the access gateway via a transport network **562**. The transport network **562** may be a wired network, wireless network, cloud network, local area network, metropolitan area network, wide area network, public network, private network, and the like.

The enterprise resources **504** may include email servers, file sharing servers, SaaS applications, Web application servers, Windows application servers, and the like. Email servers may include Exchange servers, Lotus Notes servers, and the like. File sharing servers may include ShareFile servers, and the like. SaaS applications may include Salesforce, and the like. Windows application servers may include any application server that is built to provide applications that are intended to run on a local Windows operating system, and the like. The enterprise resources **504** may be premise-based resources, cloud based resources, and the like. The enterprise resources **504** may be accessed by the mobile device **502** directly or through the access gateway **560**. The enterprise resources **504** may be accessed by the mobile device **502** via a transport network **562**. The transport network **562** may be a wired network, wireless network, cloud network, local area network, metropolitan area network, wide area network, public network, private network, and the like.

The enterprise services **508** may include authentication services **558**, threat detection services **564**, device manager services **524**, file sharing services **568**, policy manager services **570**, social integration services **572**, application controller services **574**, and the like. Authentication services **558** may include user authentication services, device authentication services, application authentication services, data authentication services and the like. Authentication services **558** may use certificates. The certificates may be stored on the mobile device **502**, by the enterprise resources **504**, and the like. The certificates stored on the mobile device **502** may be stored in an encrypted location on the mobile device, the certificate may be temporarily stored on the mobile device **502** for use at the time of authentication, and the like. Threat detection services **564** may include intrusion detection services, unauthorized access attempt detection services, and the like. Unauthorized access attempt detection services may include unauthorized attempts to access devices, applications, data, and the like. Device management services **524** may include configuration, provisioning, security, support, monitoring, reporting, and decommissioning services. File sharing services **568** may include file management services, file storage services, file collaboration services, and the like. Policy manager services **570** may include device policy manager services, application policy manager services, data policy manager services, and the like. Social integration services **572** may include contact integration services, collaboration services, integration with social networks such as Facebook, Twitter, and LinkedIn, and the like. Application controller services **574** may include management services, provisioning services, deployment services, assignment services, revocation services, wrapping services, and the like.

The enterprise mobility technical architecture **500** may include an application store **578**. The application store **578** may include unwrapped applications **580**, pre-wrapped applications **582**, and the like. Applications may be populated in the application store **578** from the application controller **574**. The application store **578** may be accessed by the mobile device **502** through the access gateway **560**, through the public Internet **548**, or the like. The application store may be provided with an intuitive and easy to use User Interface. The application store **578** may provide access to a software development kit **584**. The software development kit **584** may provide a user the capability to secure applications selected by the user by wrapping the application as described previously in this description. An application that has been wrapped using the software development kit **584** may then be made available to the mobile device **502** by populating it in the application store **578** using the application controller **574**.

The enterprise mobility technical architecture **500** may include a management and analytics capability **588**. The management and analytics capability **588** may provide information related to how resources are used, how often resources are used, and the like. Resources may include devices, applications, data, and the like. How resources are used may include which devices download which applications, which applications access which data, and the like. How often resources are used may include how often an application has been downloaded, how many times a specific set of data has been accessed by an application, and the like.

FIG. **6** is another illustrative enterprise mobility management system **600**. Some of the components of the mobility management system **500** described above with reference to FIG. **5** have been omitted for the sake of simplicity. The architecture of the system **600** depicted in FIG. **6** is similar in many respects to the architecture of the system **500** described above with reference to FIG. **5** and may include additional features not mentioned above.

In this case, the left hand side represents an enrolled mobile device **602** with a receiver **604**, which interacts with cloud gateway **606** (which includes Access Gateway and App Controller functionality) to access various enterprise resources **608** and services **609** such as Exchange, Sharepoint, PKI Resources, Kerberos Resources, Certificate Issuance service, as shown on the right hand side above. Although not specifically shown, the mobile device **602** may also interact with an enterprise application store (StoreFront) for the selection and downloading of applications.

The receiver **604** acts as the UI (user interface) intermediary for Windows apps/desktops hosted in an Enterprise data center, which are accessed using the HDX/ICA display remoting protocol. The receiver **604** also supports the installation and management of native applications on the mobile device **602**, such as native iOS or Android applications. For example, the managed applications **610** (mail, browser, wrapped application) shown in the figure above are all native applications that execute locally on the device. receiver **604** and MDX (mobile experience technology) of this architecture act to provide policy driven management capabilities and features such as connectivity and SSO (single sign on) to enterprise resources/services **608**. The receiver **604** handles primary user authentication to the enterprise, normally to Access Gateway (AG) with SSO to other cloud gateway components. The receiver **604** obtains policies from cloud gateway **606** to control the behavior of the MDX managed applications **610** on the mobile device **602**.

The Secure IPC links **612** between the native applications **610** and receiver **604** represent a management channel, which allows receiver to supply policies to be enforced by the MDX framework **614** "wrapping" each application. The IPC channel **612** also allows receiver **604** to supply credential and authentication information that enables connectivity and SSO to enterprise resources **608**. Finally the IPC channel **612** allows the MDX framework **614** to invoke user interface functions implemented by receiver **604**, such as online and offline authentication.

Communications between the receiver **604** and cloud gateway **606** are essentially an extension of the management channel from the MDX framework **614** wrapping each native managed application **610**. The MDX framework **614** requests policy information from receiver **604**, which in turn requests it from cloud gateway **606**. The MDX framework **614** requests authentication, and receiver **604** logs into the gateway services part of cloud gateway **606** (also known as NetScaler Access Gateway). receiver **604** may also call supporting services on cloud gateway **606**, which may produce input material to derive encryption keys for the local data vaults **616**, or provide client certificates which may enable direct authentication to PKI protected resources, as more fully explained below.

In more detail, the MDX Framework **614** "wraps" each managed application **610**. This may be incorporated via an explicit build step, or via a post-build processing step. The MDX Framework **614** may "pair" with receiver **604** on first launch of an application **610** to initialize the Secure IPC channel and obtain the policy for that application. The MDX Framework **614** may enforce relevant portions of the policy that apply locally, such as the receiver login dependencies and some of the containment policies that restrict how local OS services may be used, or how they may interact with the application **610**.

The MDX Framework **614** may use services provided by receiver **604** over the Secure IPC channel **612** to facilitate authentication and internal network access. Key management for the private and shared data vaults **616** (containers) may be also managed by appropriate interactions between the managed applications **610** and receiver **604**. Vaults **616** may be available only after online authentication, or may be made available after offline authentication if allowed by policy. First use of vaults **616** may require online authentication, and offline access may be limited to at most the policy refresh period before online authentication is again required.

Network access to internal resources may occur directly from individual managed applications **610** through Access Gateway **606**. The MDX Framework **614** is responsible for orchestrating the network access on behalf of each application **610**. receiver **604** may facilitate these network connections by providing suitable time limited secondary credentials obtained following online authentication. Multiple modes of network connection may be used, such as reverse web proxy connections and end-to-end VPN-style tunnels **618**.

The Mail and Browser managed applications **610** have special status and may make use of facilities that might not be generally available to arbitrary wrapped applications. For example, the Mail application may use a special background network access mechanism that allows it to access Exchange over an extended period of time without requiring a full AG logon. The Browser application may use multiple private data vaults to segregate different kinds of data.

This architecture supports the incorporation of various other security features. For example, cloud gateway **606** (including its gateway services) in some cases will not need to validate AD passwords. It can be left to the discretion of an enterprise whether an AD password is used as an authentication factor for some users in some situations. Different authentication methods may be used if a user is online or offline (i.e., connected or not connected to a network).

Step up authentication is a feature wherein cloud gateway **606** may identify managed native applications **610** that are allowed to have access to highly classified data requiring strong authentication, and ensure that access to these applications is only permitted after performing appropriate authentication, even if this means a re-authentication is required by the user after a prior weaker level of login.

Another security feature of this solution is the encryption of the data vaults **616** (containers) on the mobile device **602**. The vaults **616** may be encrypted so that all on-device data including files, databases, and configurations are protected. For on-line vaults, the keys may be stored on the server (cloud gateway **606**), and for off-line vaults, a local copy of the keys may be protected by a user password. When data is stored locally on the device **602** in the secure container **616**, it is preferred that a minimum of AES 256 encryption algorithm be utilized.

Other secure container features may also be implemented. For example, a logging feature may be included, wherein all security events happening inside an application **610** are logged and reported to the backend. Data wiping may be supported, such as if the application **610** detects tampering, associated encryption keys may be written over with random data, leaving no hint on the file system that user data was destroyed. Screenshot protection is another feature, where an application may prevent any data from being stored in screenshots. For example, the key window's hidden property may be set to YES. This may cause whatever content is currently displayed on the screen to be hidden, resulting in a blank screenshot where any content would normally reside.

Local data transfer may be prevented, such as by preventing any data from being locally transferred outside the application container, e.g., by copying it or sending it to an external application. A keyboard cache feature may operate to disable the autocorrect functionality for sensitive text fields. SSL certificate validation may be operable so the application specifically validates the server SSL certificate instead of it being stored in the keychain. An encryption key generation feature may be used such that the key used to encrypt data on the device is generated using a passphrase supplied by the user (if offline access is required). It may be XORed with another key randomly generated and stored on the server side if offline access is not required. Key Derivation functions may operate such that keys generated from the user password use KDFs (key derivation functions, notably PBKDF2) rather than creating a cryptographic hash of it. The latter makes a key susceptible to brute force or dictionary attacks.

Further, one or more initialization vectors may be used in encryption methods. An initialization vector will cause multiple copies of the same encrypted data to yield different cipher text output, preventing both replay and cryptanalytic attacks. This will also prevent an attacker from decrypting any data even with a stolen encryption key if the specific initialization vector used to encrypt the data is not known. Further, authentication then decryption may be used, wherein application data is decrypted only after the user has authenticated within the application. Another feature may relate to sensitive data in memory, which may be kept in memory (and not in disk) only when it's needed. For example, login credentials may be wiped from memory after login, and encryption keys and other data inside objective-C instance variables are not stored, as they may be easily referenced. Instead, memory may be manually allocated for these.

An inactivity timeout may be implemented, wherein after a policy-defined period of inactivity, a user session is terminated.

Data leakage from the MDX framework **614** may be prevented in other ways. For example, when an application **610** is put in the background, the memory may be cleared after a predetermined (configurable) time period. When backgrounded, a snapshot may be taken of the last displayed screen of the application to fasten the foregrounding process. The screenshot may contain confidential data and hence should be cleared.

Another security feature relates to the use of an OTP (one time password) **620** without the use of an AD (active directory) **622** password for access to one or more applications. In some cases, some users do not know (or are not permitted to know) their AD password, so these users may authenticate using an OTP **620** such as by using a hardware OTP system like SecurID (OTPs may be provided by different vendors also, such as Entrust or Gemalto). In some cases, after a user authenticates with a user ID, a text is sent to the user with an OTP **620**. In some cases, this may be implemented only for online use, with a prompt being a single field.

An offline password may be implemented for offline authentication for those applications **610** for which offline use is permitted via enterprise policy. For example, an enterprise may want StoreFront to be accessed in this manner. In this case, the receiver **604** may require the user to set a custom offline password and the AD password is not used. Cloud gateway **606** may provide policies to control and enforce password standards with respect to the minimum length, character class composition, and age of passwords, such as described by the standard Windows Server password complexity requirements, although these requirements may be modified.

Another feature relates to the enablement of a client side certificate for certain applications **610** as secondary credentials (for the purpose of accessing PKI protected web resources via the MDX micro VPN feature). For example, a work mail application may utilize such a certificate. In this case, certificate-based authentication using ActiveSync protocol may be supported, wherein a certificate from the receiver **604** may be retrieved by cloud gateway **606** and used in a keychain. Each managed application may have one associated client certificate, identified by a label that is defined in cloud gateway **606**.

Cloud gateway **606** may interact with an Enterprise special purpose web service to support the issuance of client certificates to allow relevant managed applications to authenticate to internal PKI protected resources.

The receiver **604** and the MDX Framework **614** may be enhanced to support obtaining and using client certificates for authentication to internal PKI protected network resources. More than one certificate may be supported, such as to match various levels of security and/or separation requirements. The certificates may be used by the Mail and Browser managed applications, and ultimately by arbitrary wrapped applications (provided those applications use web service style communication patterns where it is reasonable for the MDX Framework to mediate https requests).

MDX client certificate support on iOS may rely on importing a PKCS **12** BLOB (Binary Large Object) into the iOS keychain in each managed application for each period of use. MDX client certificate support may use a HTTPS implementation with private in-memory key storage. The client certificate will never be present in the iOS keychain and will not be persisted except potentially in "online-only" data value that is strongly protected.

Mutual SSL may also be implemented to provide additional security by requiring that a mobile device **602** is authenticated to the enterprise, and vice versa. Virtual smart cards for authentication to cloud gateway **606** may also be implemented.

Both limited and full Kerberos support may be additional features. The full support feature relates to an ability to do full Kerberos login to AD **622**, using an AD password or trusted client certificate, and obtain Kerberos service tickets to respond to HTTP Negotiate authentication challenges. The limited support feature relates to constrained delegation in AFEE, where AFEE supports invoking Kerberos protocol transition so it can obtain and use Kerberos service tickets (subject to constrained delegation) in response to HTTP Negotiate authentication challenges. This mechanism works in reverse web proxy (aka CVPN) mode, and when http (but not https) connections are proxied in VPN and MicroVPN mode.

Another feature relates to application container locking and wiping, which may automatically occur upon jail-break or rooting detections, and occur as a pushed command from administration console, and may include a remote wipe functionality even when an application **610** is not running

A multi-site architecture or configuration of StoreFront and App Controller may be supported that allows users to be service from one of several different locations in case of failure.

In some cases, managed applications **610** may be allowed to access a certificate and private key via an API (example OpenSSL). Trusted managed applications **610** of an enterprise may be allowed to perform specific Public Key operations with an application's client certificate and private key. Various use cases may be identified and treated accordingly, such as when an application behaves like a browser and no certificate access is required, when an application reads a certificate for "who am I," when an application uses the certificate to build a secure session token, and when an application uses private keys for digital signing of important data (e.g. transaction log) or for temporary data encryption.

Illustrative Embodiment(s)

FIGS. **7**A and **7**B illustrate sample interfaces of a mobile device, and FIGS. **8-14** illustrate sample embodiments of methods for locking a mobile device based on context. The methods depicted in FIGS. **8-14** may be combined in any suitable manner in various embodiments. The sample interfaces depicted in FIGS. **7**A and **7**B may be displayed on a mobile device, such as device **107**, **109**, **140**, **502**, and/or **602** and the methods depicted in FIGS. **8-14** may be implemented by such a mobile device.

In FIG. **8**, a flowchart of example method steps for locking a mobile device based on context is shown. The method of FIG. **8** may begin at step **802**, where a user name is logged onto a mobile device. For example, a user may log in to a mobile device, such as mobile device **502**, using a predetermined user name. The user name may be assigned by an organization, a corporation, and any suitable entity. The user name may be a portion of an organization's active directory. In an embodiment, logging on to the mobile device also requires a password, PIN, e.g., 4-8 digit number, or some other password or login credentials.

The method of FIG. **8** may proceed from step **802** to step **804**, where a context is determined for a mobile device. For example, a context may be a current time, a location of the mobile device, and other suitable contexts. The methods of FIGS. **9-12**, further described below, illustrate various embodiments where example contexts are described.

The method of FIG. **8** may proceed from step **804** to step **806**, where it is determined that the mobile device should be locked based on the user name and context. For example, the operations modes may comprise unlocked and locked. A mobile device operating in locked mode may be limited to launching a limited number of applications. For example, a plurality of applications may be presented to the user, and the user may be able to launch only the presented applications. In another example, the hardware buttons on the mobile device may be disabled. A mobile device operating in unlocked mode may be free from restrictions on launching applications and may include enabled hardware buttons. In an embodiment, the context and user name may be compared to a stored policy in order to determine an operation mode. A mobile device, such as mobile device **502**, may store one or more policies used to determine an operation mode for a mobile device. In an embodiment, the policies may be stored remotely, such as at policy manager **570**, described above with reference to FIG. **5**. For example, a user name may be logged on to the mobile device at a certain location, such as a corporate premises, and at a certain time, such as between 9:00 am and 5:00 pm. A policy may define the operation mode as locked based on the user name, location context, and current time context.

The method of FIG. **8** may proceed from step **806** to step **808**, where application to be presented to a user are selected based on the user name and password. For example, the mobile device may store a plurality of applications, and application may be selected from the stored plurality. FIG. **7**A illustrates an embodiment where user interface **700**A displayed on a mobile device (e.g., tablet, smart phone, or the like) presents Applications A **701**A, B **702**B, C **703**A, and E **704**A to a user. This is merely an example, and the plurality of applications may be presented in any suitable manner. In an embodiment, the plurality of applications may comprise email applications, web browsing applications, software-as-a-service (SaaS) access applications, and the like.

In an embodiment, the applications may be grouped based on a category. For example, a plurality of managed applications may comprise a managed group. An example of such a group may be the applications included in receiver **604**, as described above with reference to FIG. **6**. In another example, a plurality of entertainment applications may comprise an entertainment group or other genre-based groups. Any other suitable group may be implemented.

In an embodiment, a user name may be associated with one of a plurality of roles. For example, in a healthcare scenario, a role may comprise a nurse. A plurality of nursing related applications may comprise a nurse group. A policy may be defined such that an operation mode for a mobile device should be locked when a user name associated with a nurse role logs in to a mobile device at a particular location, for example, a hospital premises. The policy may also specify that the applications selected to be presented to the user are the nurse group of applications.

In a further example, a second role may comprise a doctor. A plurality of doctor related applications may comprise a doctor group. For example, a doctor group may include a prescription writing application while a nurse group does not because doctors are permitted to write prescriptions, and nurses are not. In an example, a hierarchy of roles may exist. A doctor group may be located above a nurse group in the hierarchy, and the doctor group may accordingly inherit at least all of the applications in the nurse group. A policy may be defined such that an operation mode for a mobile device should be locked when a user name associated with a doctor role logs in to a mobile device at a particular location, for example, a hospital premises. The policy may also specify that the applications selected to be presented to the user are the doctor group of applications. The converse may also be true, where a user is locked out of specific applications based on location, e.g., when a doctor logs in to a mobile device while not at the hospital premises (e.g., as determined by GPS), the springboard might present all applications but medical/work related applications.

In another example, a third role may comprise an administrator. A plurality of applications may comprise an administrator group. An administrator group may be located above both a doctor group and a nurse group in a hierarchy of roles, and the administrator group may accordingly inherit at least all of the applications in the doctor group and nurse group. A similar policy as above may be defined for a user name associated with an administrator role that logs onto a mobile device in a similar context. Alternatively, inheritance might only be partial. For example, an administrator might not inherit a prescription writing application when the Administrator is not authorized to write prescriptions under any circumstances.

The method of FIG. **8** may proceed from step **808** to step **810**, where a mobile device is locked such that a springboard presents only the selected applications. A springboard may comprise an interface that allows a user to launch applications on a mobile device. Interface **700**A of FIG. **7**A may comprise a locked springboard that presents applications A **701**A, B **702**A, C **703**A, and E **704**A. In this example, the applications presented on interface **700**A may comprise a group of applications based on a category, for example, applications for doctors. In another example, interface **700**B of FIG. **7**B may comprise a locked springboard that presents applications A **701**A, E **704**A, and G **701**B. In this example, the applications presented on interface **700**B may comprise a group of applications based on a different category, for example, applications for nurses. The mobile device may be locked on the springboard such that only the applications presented on the springboard may be launched. In an embodiment, the springboard may comprise receiver **604**, as described above with reference to FIG. **6**. In another embodiment, physical buttons on the mobile device may be disabled so that the mobile device may be effectively locked.

In an embodiment, when a presented application is launched, the user may interact with the launched application. When the user is finished and closes the launched application, the mobile device reverts back to the locked springboard. Accordingly, the user may interact only with the applications presented by the locked springboard.

In FIGS. **9-12**, flowcharts of example method steps for determining a context for a mobile device are shown. In an embodiment, steps **804**, **806**, and **808** of FIG. **8** may comprise the method steps of FIGS. **9-12**. The method of FIG. **9** may begin at step **902**, where a location for a mobile device is determined. For example, a mobile device, such as mobile device **502**, may implement the method of FIG. **9**, and a location for the mobile device may be determined. The location may be determined by GPS, signal triangulation, or any other suitable manner. The location may comprise a context for the mobile device.

The method of FIG. **9** may proceed from step **902** to step **904**, where a determined location and a user name may be compared with a location policy. For example, a policy may define, for a given user name, that a mobile device is to run in locked mode with a group of managed applications selected for presentation when the mobile device is located in a certain location, for example, on company premises. The method of FIG. **9** may proceed from step **904** to step **906**, where an operation mode is determined and applications are selected based on the comparison.

The method of FIG. **10** may begin at step **1002**, where one or more network connections are detected. For example, a mobile device, such as mobile device **502**, may implement the method of FIG. **10**, and the network connections that the mobile device makes may be detected. In an example, network connections may comprise a connection to a cellular network, a connection to a WIFI network, or a connection to a Wireless Local Area Network (WLAN), or the like. The one or more network connections may comprise a context for the selected application.

The method of FIG. **10** may proceed from step **1002** to step **1004**, where a detected network connections and a user name may be compared with a network connection policy. For example, a policy may define, for a given user name, that a mobile device is to run in locked mode with a group of managed applications selected for presentation when the mobile device is connected to a WLAN internal to a company and that a mobile device is to run in unlocked mode when the mobile device is connected to a cellular network. The method of FIG. **10** may proceed from step **1004** to step **1006**, where an operation mode is determined and applications are selected based on the comparison.

The method of FIG. **11** may begin at step **1102**, where one or more settings for a mobile device are detected. For example, a mobile device, such as mobile device **502**, may implement the method of FIG. **11**, and one or more settings for the mobile device may be detected. In an example, it may be detected whether the mobile device has a lock screen, such as a PIN required for using the mobile device, or it may be detected whether the mobile device is jailbroken, e.g., has received after-market modifications. The one or more settings may comprise a context for the mobile device.

The method of FIG. **11** may proceed from step **1102** to step **1104**, where the detected settings and a user name are compared against a settings policy. For example, a policy may define that, for a given user name, a mobile device is to run in locked mode with a group of unmanaged applications selected for presentation when the mobile device is jailbroken or lacks a lock screen.

The method of FIG. **12** may begin at step **1202**, where a current time is detected. For example, a current time may comprise the time of the day, e.g., 12:00 pm, the day of the week, a combination of these, or any other suitable timing. The current time may comprise a context for the mobile device.

The method of FIG. **12** may proceed from step **1202** to step **1204**, where the current time and a user name are compared against a current time policy. For example, a policy may define that, for a given user name, a mobile device is to run in locked mode with a group of managed applications selected for presentation when the current time is between 9:00 am and 5:00 pm on a weekday. FIGS. **9-12** describe a plurality of contexts, and any other suitable context and corresponding policy may be implemented.

In an embodiment, one or more of the contexts described in FIGS. **9-12** may be combined and these contexts, along with a user name, may be compared against a policy. For example, contexts for a selected application may comprise a location as a corporate premises and a current time as 12:00 pm on a Monday. In this example, the policy may define that, for a given user name, a mobile device is to run in locked mode with a group of managed applications selected for presentation.

In another example, contexts for a mobile device may comprise a determined location outside of the United States and a network connection with a WLAN internal to a company. A policy may define that, for a given user name, a mobile device is to run in locked mode with a group of managed applications selected for presentation. The policy may be defined in this way because a network connection with a WLAN internal to a company mitigates the risk associated with secure communications outside of the United States.

In an embodiment, the one or more contexts as described in FIGS. **9-12** may include a priority. For example, a context for a mobile device may comprise a setting as jailbroken and a policy may define that, for a given user name, a mobile device is to run in locked mode with a group of unmanaged applications selected for presentation, regardless of what other contexts indicate. Accordingly, a jailbroken mobile device will run in locked mode with a group of unmanaged applications selected for presentation even when the mobile device is connected to a WLAN internal to a company or if the selected application is located on a company premises.

In FIG. **13**, a flowchart of example method steps for navigating a locked mobile device is shown. For example, the method steps of FIG. **13** may follow the method steps of FIG. **8**. The method of FIG. **13** may begin at step **1302**, where one or more presented applications may be launched. A user may launch a presented application by interacting with a springboard, such as interface **700**A of FIG. **7**A. For example, a user may touch a presented application to launch the application.

The method of FIG. **13** may proceed from step **1302** to step **1304**, where a gesture is received as input from a user. In an embodiment, the user gesture may comprise a horizontal swipe with, for example, a finger or may comprise a vertical swipe. If the received gesture comprises a first predetermined gesture, such as a horizontal swipe, the method of FIG. **13** may proceed from step **1304** to step **1306**, where the interface switches between launched applications. If the received gesture comprises a second predetermined gesture, such as a vertical swipe, the method of FIG. **13** may proceed from step **1304** to step **1308**, where the interface switches to the locked springboard.

In an embodiment, at step **1306**, the locked mobile device only switches between launched applications that have been selected for presentation in locked mode. For example, a game application may be running on a mobile device. The mobile device may experience a change in context while running the game application that invokes a policy to lock the mobile device on a determined springboard. For example, the mobile device may enter a corporate premises and the device may be locked on a springboard that presents managed applications, and that does not present the game application. In this example, when switching applications at step **1306**, the mobile device does not allow a user to switch to the game application. This is because the game application is not part of the applications presented on the springboard, e.g., managed applications. Applications that may be switched to comprise applications presented on the locked springboard, e.g., managed applications, that have been launched.

In FIG. **14**, a flowchart of example method steps for switching an operation mode for a mobile device is shown. For example, the method steps of FIG. **14** may follow the method steps of FIG. **8**. The method of FIG. **14** may begin at step **1402**, where one or more contexts may be monitored while a mobile device is locked. In an embodiment, one or more of the contexts described with reference to FIGS. **9-13** may be monitored. For example, a locked mobile device may be located at a corporate premises at a first point in time and may then move to a new location other than the corporate premises.

The method of FIG. **14** may proceed from step **1402** to step **1404**, where a change in an operation mode for a mobile device is detected based on the monitoring. For example, a mobile device may be run in locked mode, and once the mobile device moves from a corporate premises to another location, a policy may define, for a given user name, that the operation mode for the selected application should switch to unlocked mode. The method of FIG. **14** may proceed from step **1404** to step **1406**, where the operation mode for the mobile device is switched.

Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are described as example implementations of the following claims.

What is claimed is:

1. A method comprising:

receiving a user name to log onto a mobile device;

determining a context for the mobile device based on one or more operational parameters of the mobile device;

determining, based on the context and the user name, that the mobile device is to operate in a springboard lock mode;

selecting, based on the context and the user name, one or more applications to present to a user in the springboard lock mode, wherein in the springboard lock mode, the mobile device displays a springboard user interface that presents and enables the user to access only the selected one or more applications, and wherein the user name is associated with one of a plurality of roles such that a first group of applications is selected for presentation when the user name is associated with a first role and a second group of applications is selected for presentation when the user name is associated with a second role; and

running the mobile device in the springboard lock mode.

2. A method of claim **1**, wherein determining a context comprises analyzing a location for the mobile device,

wherein when the location is on a company premises, the mobile device is determined to run in springboard lock mode.

3. A method of claim **1**, wherein determining a context comprises analyzing one or more network connections for the mobile device,

wherein when the mobile device is connected to an internal network, the mobile device is determined to run in springboard lock mode.

4. A method of claim **1**, wherein determining a context comprises analyzing a current time,

wherein when the current time is between 9:00 am and 5:00 pm on a weekday, the mobile device is determined to run in springboard lock mode.

5. A method according to claim **1**, wherein the plurality of roles is organized in a hierarchy.

6. A method according to claim **1**, wherein the selected one or more applications comprise a plurality of applications grouped by a category.

7. A method according to claim **6**, wherein the category is based on a role for the user name.

8. A method according to claim **1**, further comprising:

monitoring, while one or more applications of the selected one or more applications is running, for an updated context for the mobile device; and

switching an operating mode of the mobile device from the springboard locked mode to an unlocked mode based on the monitoring.

9. A method according to claim **1**, wherein selecting one or more applications comprises selecting, based on the context and the user name, a plurality of applications to present to the user in the springboard lock mode, and

wherein the method further comprises:

receiving, from the user via the springboard, selections of at least two applications of the plurality of applications presented on the springboard;

in response to the selections, launching the at least two applications;

receiving a predetermined gesture to switch between launched applications, wherein the mobile device only allows switching between launched applications that are presented on the springboard user interface.

10. A mobile computing device comprising:

a processor, wherein the mobile computing device is further configured to at least:

receive a user name to log onto the mobile device;

determine a context for the mobile device based on one or more operational parameters of the mobile device;

determine, based on the context and the user name, that the mobile device is to operate in a springboard lock mode;

select, based on the context and the user name, one or more applications to present to a user in the springboard lock mode, wherein in the springboard lock mode, the mobile device displays a springboard user interface that presents and enables the user to access only the selected one or more applications, and wherein the user name is associated with one of a plurality of roles such that a first group of applications is selected for presentation when the user name is associated with a first role and a second group of applications is selected for presentation when the user name is associated with a second role; and

run the mobile device in the springboard lock mode.

11. A mobile computing device of claim **10**, wherein determining a context comprises analyzing a location for the mobile device,

wherein when the location is on company premises, the mobile device is determined to run in springboard lock mode.

12. A mobile computing device of claim **10**, wherein determining a context comprises analyzing one or more network connections for the mobile device,

wherein when the mobile device is connected to an internal network, the mobile device is determined to run in springboard lock mode.

13. A mobile computing device of claim **10**, wherein determining a context comprises analyzing a current time,

wherein when the current time is between 9:00 am and 5:00 pm on a weekday, the mobile device is determined to run in springboard lock mode.

14. A mobile computing device according to claim **10**, wherein the plurality of roles is organized in a hierarchy.

15. A mobile computing device according to claim **10**, wherein the selected one or more applications comprise a plurality of applications grouped by a category.

16. A mobile computing device according to claim **15**, wherein the category is based on a role for the user name.

17. A mobile computing device according to claim **10**, wherein the mobile computing device is further configured to:

monitor, while one or more applications of the selected one or more applications is running, for an updated context for the mobile device; and

switch an operating mode of the mobile device from the springboard locked mode to an unlocked mode based on the monitoring.

18. One or more non-transitory computer readable storage media storing instructions that, when executed, configure a device to provide a user interface comprising:

a springboard user interface,

wherein the springboard user interface presents a plurality of applications to a user and enables the user to access only the plurality of applications,

wherein the plurality of applications are selected based on a context for a mobile device and a user name, wherein the user name is associated with one of a plurality of roles such that a first group of applications is selected for presentation when the user name is associated with a first role and a second group of applications is selected for presentation when the user name is associated with a second role,

wherein the context for the mobile device is based on one or more operational parameters of the mobile device,

wherein the mobile device is run in a springboard lock mode such that the mobile device is locked on the springboard user interface and the user is unable to access one or more applications on the mobile device that are inaccessible through the springboard user interface.

19. The method of claim **8**, wherein the updated context for the mobile device comprises the mobile device leaving a company premises, and wherein the switching the operating mode of the mobile device comprises in response to the mobile device leaving the company premises, switching the operating mode of the mobile device from the springboard locked mode to the unlocked mode.

20. The mobile computing device of claim **10**, wherein selecting one or more applications comprises:

selecting, based on the context and the user name, a plurality of applications to present to the user in the springboard lock mode, and

wherein the mobile computing device is further configured to:

receive, from the user via the springboard user interface, selections of at least two applications of the plurality of applications presented on the springboard user interface;

in response to the selections, launch the at least two applications; and

receive a predetermined gesture to switch between launched applications, wherein the mobile device only allows switching between launched applications that are presented on the springboard user interface.

* * * * *